l'Ordinateur de poche

Trimestriel N° 3 - 12 FF

NOUVEAUTES !

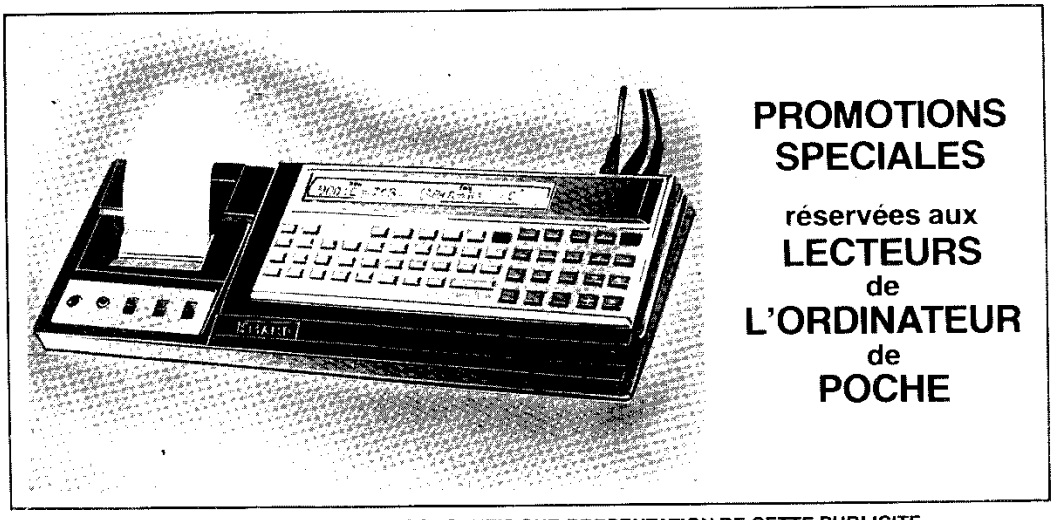
Matsushita HHC,
Hewlett-Packard 11C

Les tournées de livraison

Et toujours : initiation, jeux

M 1859-03 - 12 F

Belgique : 90 FB - Canada : 2,95 $ - Suisse : 5 FS



PROMOTIONS SPECIALES

réservées aux LECTEURS de L'ORDINATEUR de POCHE

CES TARIFS VOUS SERONT CONSENTIS SUR PRESENTATION DE CETTE PUBLICITE

SHARP	PRIX TTC
PC 1211 ORDINATEUR DE POCHE	1075,00 FRS
CE 121 INTERFACE K7	145,00 FRS
CE 122 INTERFACE K7 + IMPRIMANTE	850,00 FRS
10 RX PAPIER POUR CE 122	20,00 FRS
RUBAN ENCREUR POUR CE 122	20,00 FRS
10 CASSETTES VIERGES C15	70,00 FRS

VICTOR-LAMBDA	PRIX TTC
VICTOR V1 16K + PERITEL	3950,00 FRS
MANETTES DE JEU (L'UNITE)	130,00 FRS
PROGRAMMES DIVERS A PARTIR DE	99,00 FRS

VIDEO GENIE SYSTEM	PRIX TTC
VGS EG 3003 CLAVIER CPU + CASSETTE	4150,00 FRS
VGS EG 3008 CLAVIER CPU + CLAVIER NUM	4590,00 FRS
VGS EG 3014 BOITIER EXPANDER	3060,00 FRS
VGS EG 400 SIMPLE DRIVE	3250,00 FRS
VGS EG 101 MONITEUR VERT 12"	1120,00 FRS
VGS CABLE LIAISON DRIVE	280,00 FRS

ACCESSOIRES DIVERS	PRIX TTC
10 K7 C15 SCOTCH 3M	70,00 FRS
MAGNETO K7 AVEC SACOCHE (SECTEUR/PILES)	380,00 FRS
10 DISQUETTES VERBATIM	290,00 FRS

PROMOTION DU MOIS

SHARP PC 1211 + CE 122 + 10 Rx papiers / CE 122 **1900,00 TTC**

J.C.R. Electronique - 58, rue Notre-Dame-de-Lorette - 75009 Paris - Tél. (1) 282.19.80

Expédition dans toute la France - Matériel garanti 1 an pièces et m.-o. - Ouvert du mardi au samedi de 10 h à 13 h et de 14 h à 19 h

En raison des fluctuations monétaires, ces prix sont susceptibles d'être modifiés sans préavis. Nous consulter pour confirmation.

1
COUVERTURE
Comme on le voit, même dans la cité du futur les micropoches sont à leur place dans le décor imaginé par *Jean-Paul Chagniot*

5
ÉDITORIAL

7
À VOS CLAVIERS

12
MAGAZINE

16
DES ORDINATEURS DE POCHE POUR...
organiser les tournées de livraison d'une grande entreprise : des poids-plume aident à résoudre les problèmes posés par les poids-lourds.

19
N'AVEZ-VOUS RIEN OUBLIÉ ?
La mémoire de votre micropoche est fidèle. Pouvez-vous en dire autant de la vôtre ?

20
UN PROGRAMME DÉCORTIQUÉ
Le Jeu de la Vie adapté pour PC 1211/TRS 80 Pocket avec des explications particulièrement détaillées.

25
BASIC OU NON ?
Quelques réflexions sur les avantages respectifs des différents langages de programmation.

27
JOUEZ AU TIERCÉ ET AU LOTO
Un programme pour simuler ces deux jeux sur une TI 59.

29
CALCUL MENTAL
Dans le domaine des quatre opérations, le micropoche a réponse à tout, mais il peut aussi nous poser des questions. Exerçons-nous.

32
NOUVEAU : LE HEWLETT-PACKARD 11 C
Un premier aperçu de ce micropoche à orientation scientifique.

34
NOUVEAU : CHEZ MATSUSHITA, LE HHC QUASAR
Attendu depuis longtemps, ce micropoche arrive avec son BASIC.

37
LE LASER D'OMÉGA
Un jeu d'anticipation pour Casio FX 502 P dont l'afficheur devient l'écran du radar du vaisseau spatial Oméga.

l'ordinateur de poche

n° 3

12 F - Trimestriel

RÉDACTION-RÉALISATION

Rédacteur en chef : Bernard Savonet
Rédacteurs : Jean-Baptiste Comiti, Jean-Pierre Brunerie
Assistante de rédaction : Michelle Aubry
Maquette : Eliane Gueylard
Ont participé à ce numéro :
Denis Alexandre, Arnaud Beauregard, Paulette Besnard, Didier Bicking, Dominique Coupleux, Micheline Domancich, Jean Drano, Dominique Fibo, Bruno Garreau, Jean-Christophe Janin, Xavier de La Tullaye, Eric Martin, Robert Pulluard, Yves Roque, Christophe Théron, Jean Thiberge, Daniel-Louis Tipule, Antoine Vésigot, André Warusfel.
Iconographie :
Eric Berthier, Serge Fayol, Nicolas Spinga.

ÉDITION-PUBLICITÉ
Editeur : Jean-Pierre Nizard
Assistante d'édition : Maryse Marti

Rédaction-vente-publicité : 41 rue de la Grange-aux-Belles, 75483 PARIS CEDEX 10. Téléphone : (1) 238.66.10 - Télex : 230 589 EDITEST.
Abonnement voir page 6

l'Ordinateur de poche est une publication du *groupe* tests
Directeur de la publication : Jean-Luc Verhoye

40
ÉCOUTEZ VOTRE MICROPOCHE
Le PC 1211/TRS 80 Pocket peut se faire entendre grâce à un récepteur radio : une drôle de musique.

41
LE BON ANGLE DE TIR
Un programme pour TI 57 — où vous jouer le rôle d'un serveur de DCA.

42
RÉTABLIR L'ORDRE ALPHABÉTIQUE
Pour votre HP 41 C, un joli casse-tête inspiré par du fameux cube de Rubik.

45
INITIATION
Une leçon où l'on étudie sur TI 57 comment utiliser les boucles de programmes.

50
ENCORE UNE CARTE ?
Le micropoche les tire au hasard. Comptez bien vos points : vous ne devez pas dépasser 13.

51
LES INSTRUCTIONS LOGIQUES
Le BASIC comporte des fonctions logiques : regardons-les de plus près.

55
CONSULTEZ LA BASE
Le système décimal, qui nous est si familier, n'est qu'un système parmi d'autres. On peut en changer, mais comment faire ?

57
DANS UN FAUTEUIL
Le confort, en informatique de poche, passe parfois par un bricolage très banal (prévoyez un marteau).

60
TIREZ LE MAXIMUM
Les micropoches peuvent résoudre des problèmes de mathématiques qui sont aussi des sujets d'examen.

63
UN POT COMMUN POUR TOUTES LES MACHINES
Pour votre TI 57 et votre HP 33, les versions de quelques programmes.

66
UN ANCÊTRE
Le *Compucorp Scientist* aura bientôt dix ans. C'était l'une des premières calculatrices programmables de poche. Honneur aux anciens !

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemples et d'illustrations, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause est illicite » (alinéa 1er de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre façon sanctionnée par les Art. 425 et suivants du Code Pénal.

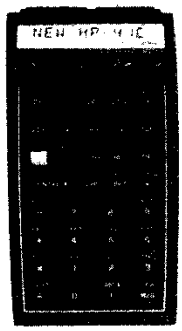
ILLEL

ESPACE ELECTRONIQUE

MICRO-INFORMATIQUE/HI-FI/VIDÉO

ILLEL CENTER PARIS 10e, OUVERTURE LE 1er SEPT. 81/650 M²/3 NIVEAUX
86, bd Magenta. Tél. 201 94 68. Métro : Gare du Nord, Gare de l'Est. Parking Magenta
ILLEL CENTER PARIS 15e, 143, av. Félix-Faure. Tél. 554 97 48. Métro : Balard
Nos magasins sont ouverts de 9 h 30 à 12 h 30 et de 14 h à 19 h. Le lundi à partir de 15 h

OFFRE SPÉCIALE
Quantité limitée.



Hewlett-Packard HP-41 C

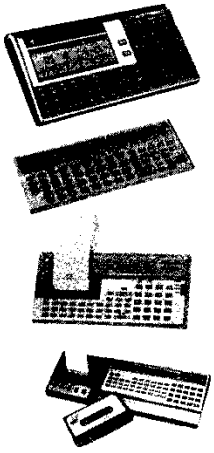
La calculatrice HP 41 est apte à converser en alphanumérique donc vous pouvez dialoguer avec le calculateur dans votre langue, et appeler les programmes par leur nom. Des indicateurs vous informent en permanence du mode d'utilisation et de l'état de la batterie.

Les deux calculateurs HP-41 sont entièrement programmables. La mémoire permanente du HP-41 CV offre, soit un maximum de 2.000 lignes de programmes, soit 319 registres de données, soit toute combinaison de lignes et de registres. Par contre, la mémoire permanente du HP-41 C propose 400 lignes de programmes, 63 registres de données ou toute combinaison des deux. Des modules permettent de passer de la mémoire du HP-41 à celle du HP-41 CV.

Le HP 41 est un calculateur souple, sur mesure, pouvant relier un calculateur préprogrammé à un instrument adapté à vos besoins.

1790 F
Prix au comptant TTC

une nouvelle génération les ordinateurs de poche



EL 6200
Double affichage à cristaux liquides par points.
Affiche la date (année, mois, jour); l'heure (heures, minutes) et le mémorandum des messages et rendez-vous.
Permet d'identifier les appels téléphoniques, réunions, etc.
Programme journalier et mensuel sur simple presse-bouton.
Montre avec réveil. Calculatrice avec mémoire.

EL 5100
Des formules entières peuvent être introduites à l'aide de la fonction "Direct Formula Entry". Les formules algébriques compliquées peuvent être stockées par la fonction de mise en réserve d'expressions algébriques.
Fonction de lecture. Mémoire de données multiples. 61 fonctions.

EL 7000
Impression de caractères et de chiffres sur rouleau de papier normal.
Méthode et résultat des calculs imprimés avec bref commentaire.
Mémoire à quatre touches. Mémoire à 8 mots.
Mémoire de caractères jusqu'à 120 signes.
Fonction d'arrêt automatique. Protection de la mémoire.

PC 1211
Programmable en Basic1

HEWLETT-PACKARD	T.T.C.
HP-41 CV	2.350,00
LECTEUR DE CARTE	1.390,00
IMPRIMANTE THERMIQUE	2.490,00
LECTEUR DE CODES BARRES	890,00
MODULE QUADRUPLÉ	750,00
MODULE MÉMOIRE PROGR.	240,00
BATTERIE + PORTE BATT.	120,00
CHARGEUR	240,00
BIB. STATISTIQUES	240,00
BIB. MATHS.	240,00
BIB. FINANCIÈRES	240,00
BIB. JEUX	240,00

SHARP	T.T.C.
PC 1211	nous consulter
CE 121	200,00
CE 122 (impri.)	970,00
EL 7000	770,00
EL 6200	690,00
EL 5100	670,00
IQ 3100	1.090,00
IQ 31 F MOD. FRANÇAIS	280,00
IQ 31 G MOD. ALLEMAND	280,00
IQ 31 S MOD. ESPAGNOL	280,00
40 CARTES MAGNETIQUES	180,00
3 BOÎTES DE 40 CARTES	360,00
BOÎTE DE 6 ROULEAUX PAP	48,00
MANUEL D'UTILISATION	120,00

Propositions d'Ensembles

1 PC 1211	1 PC 1211	1 HP - 41 C
1 CE 121	1 CE 122 imprimante	1 lecteur de cartes
1 magnétophone	1 magnétophone	1 batterie + support
5 cassettes vierges	5 cassettes vierges	1 chargeur
1.690,00 F TTC	**2.590,00 F** TTC	**3.390,00 F** TTC

Liste générale de nos micro-ordinateurs et périphériques :
APPLE – COMMODORE – HEWLETT-PACKARD – SHARP – SANCO-SANYO – MICROSOFT – CALIFORNIA COMPUTER SYSTEM – CENTRONICS – MICROLINE – EPSON – AXIOM – TRENDCOM – QUME – SANYO – MOUNTAIN HARDWARE – VERBATIM – DYSAN

à découper, à remplir et à retourner à ILLEL CENTER INFORMATIQUE service vente par correspondance 143, avenue Félix-Faure, 75015 Paris.

Je commande ferme et désire recevoir en urgence le matériel suivant : _____

au prix HT de F _____ + TVA 17,60 % _____ = TOTAL TTC _____ N° téléphone _____

Mode de règlement : Comptant ☐ Crédit* ☐ Leasing** ☐ Je verse au comptant la somme de (20 % minimum pour le crédit) _____ F

Ci-joint : Chèque bancaire ☐ CCP ☐ Mandat-carte ☐ NOM _____ PRÉNOM _____

ADRESSE _____ CODE POSTAL _____

* Conditions de crédit : CREG ● être salarié, ● 20 % minimum au comptant, solde arrondi à la centaine supérieure.

** Conditions de leasing : SOVACREG ● être salarié, ● pas de versement comptant, loyer réparti sur 48 ou 36 mois.

Date et signature :

Ajouter 30 F de port et d'emballage pour toute commande inférieure à 2 000 F TTC (pas d'envoi contre-remboursement)

Le micro dans la poche

Nous recevons encore des réponses au questionnaire " lecteur qui êtes-vous " que nous vous proposions dans notre dernier numéro, mais pour avoir dès ce numéro-ci les résultats de la question " xxxpoche ", nous avons fait un dépouillement partiel sur quelque 1300 bulletins qui nous étaient parvenus fin août. Bien que partiels, ces résultats peuvent en fait être considérés comme définitifs car, sauf afflux massif de bulletins identiques entre eux, il ne devrait pas y avoir de variations suffisantes pour modifier profondément l'issue de ce sondage que nous arrêtons le 25 septembre.

Alors, finalement, quel est le mot que *vous* avez choisi ? Incontestablement, c'est **micropoche** qui arrive en tête. Il est suivi d'assez près par *ordipoche*, et d'un peu plus loin par l'abréviation *Op*. Notre favori de dernière minute (oui, nous en avions un : c'était *poquette* !) ne s'est, hélas pour notre amour-propre, classé qu'à la vingt-et-unième place sur 26 : pas fameux. A partir de maintenant, c'est donc *micropoche* que nous nous efforcerons d'adopter systématiquement. Certains articles de ce numéro étaient déjà terminés avant l'obtention du résultat, aussi ne nous en veuillez pas trop des quelques *poquettes* qui nous ont échappé !

Les autres informations pour lesquelles nous avons effectué un dépouillement provisoire sont les modèles de machines. Nous ne donnerons pas de chiffres dont la précision apparente serait trompeuse, car notre enquête ne répond pas *pour cette question* aux critères exigés d'un sondage public : il est en effet impossible d'assurer que l'échantillon des lecteurs qui ont répondu est, comme disent les spécialistees, *représentatif* de l'ensemble de nos lecteurs, et bien entendu encore moins de l'ensemble de la population.

Il semble toutefois s'en dégager que vous avez en général plusieurs machines. Ce sont bien sûr les TI les plus nombreuses, et de très loin, devant les HP. Le Sharp PC 1211 et son cousin le TRS-80 poquette (oh pardon !) suivent ensuite, performance tout à fait remarquable puisqu'il s'agit d'un modèle unique par rapport aux deux familles des " grands " : ils dépassent même la HP-41C/CV dont les possesseurs ont dû se montrer vraiment paresseux pour répondre à notre enquête. La pauvre Casio FX-502P manifeste par contre dans vos réponses une discrétion remarquablement inquiétante et relativement imméritée : gageons qu'il n'en sera pas de même pour le nouveau micropoche (là, j'y ai pensé !) de la marque, le 702.

Car bien sûr, les résultats ci-dessus ne sont qu'un instantané de la situation actuelle. L'arrivée des nouveaux micropoches, tels ceux que nous vous présentons dans ce numéro, jouera certainement dans l'évolution des prochains mois.

En attendant, il ne nous reste plus qu'à effectuer les tirages au sort que nous vous avons promis pour les abonnements à vie. Au moment où vous lirez ce numéro, ce sera chose faite, surtout si nous sortons en retard : le tirage se fera en effet à la Boutique Informatique pendant le Sicob. Peut-être même, si vous êtes gagnant, aurez-vous déjà reçu notre lettre vous l'annonçant...

☐ Bernard Savonet

ABONNEZ-VOUS A

l'Ordinateur de poche

C'est la seule revue française exclusivement dédiée à l'informatique de poche.

Si vous possédez déjà une calculatrice programmable, vous trouverez dans L'ORDINATEUR DE POCHE des tas d'astuces qui vous permettront de tirer un meilleur parti de votre machine.

Si vous envisagez d'en acheter une, L'ORDINATEUR DE POCHE sera pour vous un guide de choix irremplaçable.

N'hésitez pas à investir 45 Francs(*) pour une meilleure information.

Retournez aujourd'hui même le bulletin d'abonnement ci-dessous.

Bulletin à retourner à
L'ORDINATEUR DE POCHE Service Abonnements
41, rue de la Grange aux Belles 75483 Paris Cedex 10

Nom _____ Prénom _____

Adresse _____

Pays _____ Code postal _____ Ville _____

Veuillez m'abonner pour un an (4 numéros) à L'ORDINATEUR DE POCHE,
ci-joint mon règlement de 45 FF.
(*Etranger : 60 FF, Belgique : 320 FB, Suisse : 18 FS).

A vos claviers

Un scaphandre pour le PC 1211

A l'attention de Jean-Pierre Azra.

Pour protéger une programmette de la poussière de craie, il suffit de l'enfermer dans un emballage de coton pharmaceutique (1). C'est à la fois étanche et transparent, ça a la dimension requise, et c'est suffisamment souple pour qu'on puisse utiliser l'ordinateur à travers sa protection.

Il y a également la solution des boîtiers étanches pour appareils photographiques mais elle est plus coûteuse, plus encombrante et moins commode.

Paul Canal
78 Versailles

(1) Je conseillerais en particulier les sacs de coton de la marque « Wilot » (vente exclusive en pharmacie)... Publicité non payée !

Déjà deux anciens numéros

Après avoir longuement cherché votre premier numéro, je n'ai pu obtenir que le second. Serait-il possible de vous commander ce premier *Op ?*

Il me serait en effet agréable d'avoir tous les numéros, d'autant plus que je possède une TI 58C et que ce premier magazine présentait des programmes écrits pour ma machine.

Vincent Gaillon
37 Saint-Pierre-des-Corps

Ayant découvert la revue *l'Op,* je me suis jeté dessus, j'ai plongé dedans, et j'ai littéralement dévoré à pleines dents le programme de l'avion-espion pour la TI 57 (que je possède).

Seulement voilà : je n'arrive pas à trouver *l'Op* n° 1. J'ai vainement cherché dans différentes librairies. Je voudrais donc connaître le moyen d'acquérir ce n° 1.

Albert Van Lottum
Grace-Hollogne, Belgique

Je tiens à vous féliciter pour votre revue que je n'ai malheureusement découverte qu'à son 2e numéro. Je n'ai pas réussi à trouver le premier numéro en Suisse Romande. Pourriez-vous me dire s'il est encore disponible et de quelle façon je peux me le procurer. Je m'abonnerais volontiers, mais j'ai un ami qui a un magasin et qui se fait un plaisir de me réserver les numéros suivants.

A. Murguerat
Paudez, Suisse

■ *Que ce soit en Suisse, en Belgique, au Canada ou en France, il vous est possible de commander les anciens numéros de* l'Op *à votre marchand de journaux.*

Vous pouvez également adresser votre commande accompagnée de votre règlement à l'Ordinateur de poche, *service numéros.*

Suisse :
27 route du Grand-Mont
CH 1052 Le Mont-sur-Lausanne

Belgique :
204 avenue Brugman
1180 Bruxelles

Canada :
LMPI
4435 bd des Grandes-Prairies, Montréal
Québec H1R 3N4

France et autres pays :
41 rue de la Grange-aux-Belles
75483 Paris Cedex 10

Le prix d'un numéro antérieur est le même que celui du dernier numéro, à savoir 12 FF (pour la France), 5 FS (Suisse), 90 FB (Belgique), 2,95 $ (Canada) et 16 FF pour les autres pays. Pour éviter toute complication, nous vous demandons de ne jamais utiliser le système du contre-remboursement. Merci.

Croqué sur le vif

Depuis le début de juillet, j'interrogeais chaque jour ma marchande de journaux — à la fin, c'était du regard seulement — et toujours « non » !

Je voyais avec anxiété approcher la date du départ en vacances et me demandais, tel Hamlet : « Partir sans le n° 2 de *l'Op* ou ne pas partir ? ». To go or not to go, c'est le monde à l'envers pour un basicien.

Et puis, hier soir, à ma descente de train, elle m'a fait de grands signes : « *Il* est arrivé, et comme je n'en ai reçu qu'un, je vous l'ai gardé ! » — gentil, non ? La largeur du comptoir m'a empêché de la remercier comme elle le méritait, mais ce n'est que partie remise.

Du coup, je m'empresse de m'abonner, de peur qu'un moment d'inattention au mois d'octobre ne me prive du n° 3.

Amicalement

M. Mortier
06 Cap d'Antibes

Le registre T comme TI

Le programme Avion-espion pour TI 57 présenté dans le n° 1 devrait pouvoir être retranscrit facilement pour TI 59. Or ce n'est pas le cas.

. La séquence des instructions aux pas 21, 22 et 23 est-elle bien juste ?

. Si C.t efface le registre t, à quoi correspond INV C.t au pas 38 ?

E.A. Abaglo
Lomé, Togo

■ *Sur la TI 57, la touche C.t, plus précisément la séquence 2nd C.t efface bien, comme vous l'indiquez, le registre t, registre t qui n'est autre que la mémoire n° 7. Quant à la séquence INV 2nd C.t, elle fait exactement l'inverse : elle efface toutes les mémoires,* ***sauf*** *la mémoire t.*

Sur les TI 58, 58C et 59 qui sont pourtant de la même famille que la TI 57 les choses se passent de façon légèrement différente, puisque le registre t est un registre indépendant. Pour y ranger une valeur, on commence par la placer à l'affichage (registre x) et l'on exécute, manuellement ou par programme x ⇄ t, c'est-à-dire « x échange t ». L'ancien contenu de t se retrouve à l'affichage, et le nombre qui était à l'affichage est stocké dans le registre t.

Pour effacer le contenu de la mémoire t, il suffit d'inscrire 0 à l'affichage et d'effectuer x ⇄ t.

La séquence 2nd CMs (effacement des mémoires) efface toutes les mémoires de données de la machine ***sauf*** *le registre t. C'est donc elle qui correspond à l'INV 2nd C.t de la TI 57.*

Se perdre en conjectures...

J'ai été étonné de trouver, page 20 de votre deuxième numéro, une question non résolue par les grands mathématiciens et accessible, en apparence, aux calculettes, mais (hélas) pas aux esprits tels que le mien ! Combien y a-t-il de questions posées, non encore résolues par le cercle des scientifiques de toute discipline, et qui, pourquoi pas, pourraient bien avancer vers une solution si quelqu'un rend publiques ces interrogations à un vaste public tel celui de *l'Op ?*

Bernard Baruton
31 Toulouse

■ *Les conjectures mathématiques sont des propositions qui n'ont pas encore reçu de démonstration et dont personne n'a pu prouver qu'elles étaient fausses. On ne sait donc pas, à proprement parler, si elles sont vraies ou fausses.*

Dès qu'une de ces propositions a été rigoureusement démontrée, c'est-à-dire lorsqu'on a pu prouver qu'elle était toujours vraie, elle devient théorème. Par contre, s'il se trouve quelqu'un pour montrer d'une façon ou d'une autre que la conjecture n'est pas toujours vraie, elle cesse d'être conjecture, et il devient bien entendu inutile d'essayer de lui trouver une démonstration : la conjecture était *fausse.*

Tant que la démonstration n'a pas été apportée de façon irréfutable, on peut appliquer la conjecture à des cas particuliers et vérifier qu'elle est juste pour tous ces cas. Comme le nombre de cas possibles est généralement infini, cette approche du problème ne peut pas conduire à une démonstration, mais on peut tout de même s'y employer avec l'espoir de trouver un cas qui prenne la conjecture en défaut : on démontrerait alors qu'elle est fausse, car en mathématique, les exceptions ne confirment pas les règles...

Dans cette recherche, les ordinateurs sont des outils très efficaces principalement grâce à leur vitesse de calcul. Avec un programme de quelques lignes et un ordinateur individuel, on peut par exemple s'assurer que la conjecture tchèque est vérifiée pour tous les nombres compris entre 1 et 10 000 en moins d'une nuit. Si quelqu'un s'avisait de faire ce travail à la main sérieusement, on peut supposer qu'il y passerait toutes ses vacances, en pure perte d'ailleurs, car on peut également parier que sur les centaines de milliers d'opérations qu'il aurait à faire, il commettrait quelques erreurs.

Quand on sait qu'un ordinateur de poche vous donnera en quelques semaines plus de nombres premiers que vous ne pourriez en trouver vous-même si vous décidiez d'y consacrer toute votre vie, on comprend pourquoi les personnes qui ont fait de l'arithmétique leur passe-temps sont dans le ravissement lorsqu'elles découvrent l'informatique.

Il existe beaucoup de conjectures célèbres. Certaines sont de véritables casse-tête : on a l'impression que la démonstration ne devrait pas être si difficile que cela et à mesure qu'on avance, on découvre que le problème est plus fuyant qu'on ne l'avait pensé. Si l'on se pique au jeu, on peut même y passer tous ses loisirs.

En voici une qui attend sa démonstration depuis plus d'un siècle ; c'est celle dite de Goldbach, et elle s'énonce ainsi : « tout nombre entier est la somme d'au plus trois nombres premiers ». Ce qui peut se dire également « tout nombre pair est la somme de deux nombres premiers ». A vos machines.

A vos claviers

Un programme très demandé

Le programme de conversions publié dans *l'Op n° 2* pourrait rendre service à de nombreux médecins ou internes, ou étudiants en médecine.

Beaucoup sont comme moi possesseurs non d'une TI 57, mais d'une HP 41C. Beaucoup sont peut-être aussi débutants. Y a-t-il un astucieux ou courageux collaborateur pour transformer le programme Missirliu TI 57 en un programme HP 41C ?

A vos claviers secourables !

Jean Canet
94 Créteil

■ *Parions qu'il se trouve parmi les médecins et les biologistes plusieurs utilisateurs de la HP 41C qui ont déjà adapté ce programme sur leur ordinateur de poche, et il est certain qu'ils ne le garderont pas pour eux. Nous attendons avec confiance leur courrier.*

Ne vous envolez pas vers les sommets

Juste un petit mot pour vous féliciter de votre trimestriel (trimestriel à quelques semaines près), il reste très intéressant même s'il est parfois difficile de se le procurer. Une seule solution : s'abonner ! C'est fait. Les articles me plaisent, mais de grâce ne vous envolez pas vers les cieux où règnent les grosses têtes de l'informatique. Sinon, de nombreux lecteurs ne pourront plus suivre. (Cette critique est formulée à titre préventif car pour l'instant la revue est très homogène.)

M. Levy
95 Franconville

Je dois vous signaler une erreur dans le programme BASIC du « jeu de dés » (n° 2, page 31). A la ligne 410, il est imprimé IF Ó RETURN (je l'ai rapidement corrigée moi-même mais cela pourrait décourager un débutant !).

Pierre-Yves Descamps
Bruxelles, Belgique

J'ai rencontré un problème lors de la lecture de programmes pour PC 1211. On trouve parfois des lignes telles que
IF Ó RETURN ou IF D LET A = 54

Que signifient ces lignes, quel en est l'effet et comment les traduire sur un plus gros ordinateur individuel ?

François Torreilles
63 Durtol

■ *Les fonctions logiques des différentes versions du BASIC sont — il ne manquerait plus que cela ! — logiques.*

Ce sont par ailleurs des outils très puissants pour la programmation. Sur le BASIC du TRS 80 Pocket et du Sharp PC 1211, ces fonctions existent bel et bien, et nous publions dans le présent numéro un article qui leur est consacré.

Vous y trouverez les éléments qui vous permettront de comprendre la façon dont elles fonctionnent et les services qu'elles peuvent rendre.

Noir sur blanc, SVP

Tous les programmes que vous éditez pour PC 1211 Sharp sont très difficilement lisibles. Par exemple en page 51 du n°2, quel est le caractère contenu dans la première parenthèse de la ligne 400 ? Illisible ! Idem pour les lignes 55, 70, 90, 100, 120, etc... de la page 61.

Cela est dû au fait que la couleur violette de l'encre de l'imprimante passe très mal à l'imprimerie du journal comme à la photocopie d'ailleurs.

Dans le précédent numéro, vous aviez fait par contre un programme parfaitement lisible pour le « nombre secret » sur PC 1211.

Il faut absolument corriger cela, car ces programmes sont très intéressants.

J.-M. Bilbille
18 Mehun-sur-Yèvre

J'ai une critique à formuler sur le journal : pourquoi les listings des programmes pour PC 1211 - TRS 80 Pocket sont-ils aussi mal imprimés ? Les autres listings (TI 59, HP 41C) sont très lisibles, alors d'où vient la différence ?

Jean Thioulouse
69 Lyon

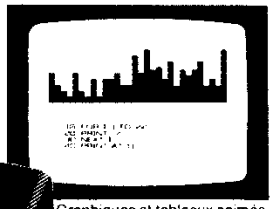
Dans le lycée que je fréquente, nous sommes plusieurs à suivre avec intérêt vos articles (...) Une seule ombre au tableau : la publication des programmes. Il nous a été très difficile de lire le programme sur PC 1211 du numéro 2 page 61.

Philippe Rocca
06 Nice

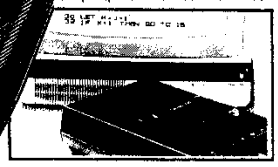
Micro-ordinateur individuel Sinclair ZX 81

985F TTC

Manuel gratuit, prise secteur gratuite, TVA et frais d'envoi compris.



Graphiques et tableaux animés.



Utilisez votre propre téléviseur comme moniteur et votre propre magnétophone pour conserver votre programme.

Cours gratuit de programmation en Basic sans expérience préalable nécessaire (en français).

En une journée vous lui parlerez comme à un vieil ami.

Si les ordinateurs vous intéressent vous trouverez le ZX81 tout à fait fascinant.

Plus encore, vous le trouverez d'une grande utilité. La connaissance de l'ordinateur qu'il vous donnera vous sera utile aussi bien dans les domaines professionnels que familiaux. L'apprentissage donné à vos enfants leur rendra service toute la vie.

Le ZX81 élimine les mythes de l'informatique, il vous enseigne immédiatement le Basic, le plus simple des langages informatiques.

Vous prenez le ZX81, vous le branchez à votre téléviseur et vous êtes prêt à commencer.

Manuel en main vous monterez un programme en une heure ; en une semaine vous élaborerez vos propres programmes complexes avec confiance et compétence, et cela pour moins de 1000 francs.

Les caractéristiques du ZX81, parfaites pour l'apprentissage, le rendent facile à l'utilisation.
Le ZX81 est fait pour faciliter l'étude des débutants, il permet d'incorporer les ordres de façon simple et claire et d'identifier les erreurs et même de les corriger sans difficultés.

Mais cela ne signifie pas qu'il soit seulement un jeu. Le ZX81 est un ordinateur très rapide et très puissant tout à fait capable de donner des résultats que vous comparerez aux ordinateurs individuels plus grands et plus chers.

Comment peut-on offrir un vrai ordinateur pour seulement 985 francs ?
Tout simplement, à la base, nous avons utilisé l'ordinateur conventionnel et l'avons créé en 4 circuits concentrés d'une puissance extrême dont l'un est le circuit maître, unique et particulier. Le résultat n'est pas seulement une machine mais le cœur même de l'informatique.

Au fur et à mesure que vos connaissances et vos besoins se développeront, votre ZX81 les suivra si vous ajoutez 16 fois plus de capacité de mémoire avec le Sinclair 16 K-octets RAM.

L'imprimante propre au Sinclair vous offre toutes les caractéristiques alpha numériques et des fonctions graphiques évoluées.

Vous recevrez les détails des Soft ZX (programmes déjà enregistrés sur cassettes) avec le ZX81.

Votre cours d'informatique ZX81 en français.
Le ZX81 est accompagné d'un nouveau guide de l'informatique de plus de 200 pages. Ce manuel ne nécessite aucune connaissance préalable et présente un cours complet des principes de base aux programmes les plus complexes.

Sa structure équilibrée entre la théorie et la pratique vous permet non seulement de lire mais de créer, ce qui rend l'apprentissage facile et plaisant.

Le prix T.T.C. comprend :
Le transformateur, les câbles (connecteur TV et enregistreur à cassettes), le port et l'emballage.

Votre téléviseur vous servira de moniteur sans que la réception ultérieure en soit affectée.

Le ZX81 est accompagné des câbles et prises nécessaires au branchement à tout téléviseur familial en noir ou en couleurs.

Si vous choisissez d'utiliser un enregistreur à cassettes pour conserver les programmes, il vous suffira d'utiliser des cassettes normales.

Si vous avez un ZX80 :
Nous vous indiquerons par téléphone au 359.72.50 ce qu'il y a lieu de faire pour augmenter ses capacités.

Spécifications du ZX81 :

- Microprocesseur, version nouvelle et plus rapide du microprocesseur de réputation mondiale.
- Fonction d'entrées du mot clef par une touche unique éliminant un travail fastidieux.
- Contrôle inédit de la syntaxe identifiant immédiatement les erreurs de programmation.
- Une gamme complète de fonctions mathématiques et scientifiques à 8 décimales.
- Traçage de graphiques et tableaux animés.
- Tableaux numériques et chaîne multi-dimensionnelle.
- Boucles imbriquées FOR/NEXT (26 maximum).
- Fonction randomisée pour les jeux aussi bien que pour les applications plus sérieuses.
- Cassette Load et Save titrés.

Comment commander votre ZX81.

- Par poste, utiliser le coupon ci-dessous.
- Votre commande vous parviendra sous 4 semaines et nous vous donnons la possibilité de nous le retourner dans un délai de 15 jours après réception et vous serez remboursé intégralement.
  Nous désirons vous donner entière satisfaction et tout sera fait pour y parvenir.

sinclair
ZX81

Découpez ce bon et envoyez-le à :
DIRECO INTERNATIONAL,
30, avenue de Messine. 75008 PARIS. 359.72.50

Je désire recevoir sous 4 semaines, par paquet-poste recommandé :
☐ le micro-ordinateur Sinclair ZX 81 avec son adaptateur secteur et le manuel BASIC pour le prix de 985 F T.T.C.
☐ l'extension de mémoire RAM (16 K-octets) pour le prix de 650 F T.T.C.
☐ l'imprimante pour le prix de 690 F T.T.C.

Je choisis de payer :
☐ par C.C.P. ou chèque bancaire établi à l'ordre de DIRECO INTERNATIONAL, joint au présent bon de commande.
☐ directement au facteur, moyennant une taxe de contre-remboursement de 14 F.

Nom _____
Prénom _____ N° _____
Rue ou Lieu-dit _____
Commune _____ Code postal _____
Localité du bureau de poste _____
(Pour les moins de 18 ans, signature de l'un des parents.) Signature.

OP 1

■ *Comme vous avez, hélas, pu vous en apercevoir, il n'est pas facile de parvenir à ce que la liste d'un programme PC 1211/TRS 80 Pocket soit reproduite de façon lisible dans les pages de votre journal.*

La première chose qu'il faut avoir bien présente à l'esprit, c'est que l'on a tout intérêt à insérer dans la revue la liste du programme telle qu'elle est sortie de l'imprimante CE-122 ou de celle que commercialise Tandy pour son TRS 80 Pocket : on est ainsi certain (ou presque) qu'aucune erreur de transcription ne se glissera entre le programme qui a tourné *dans la machine et le programme publié.*

Il faut également savoir que, pendant sa fabrication, le journal connaît une phase de photographie au cours de laquelle certains tons de bleu ne sont pas reproduits : tout se passe comme si la pellicule était aveugle à cette couleur, et il en va de même pour certaines photocopieuses. Ceux d'entre vous qui se sont essayés à photocopier leurs propres listes l'auront peut-être constaté par eux-mêmes : lorsqu'on essaie de reproduire une liste obtenue grâce à l'imprimante de chez Sharp, il arrive que la photocopieuse n'y voie que du bleu ! En tout cas, le moindre qu'on puisse dire est que le rendu *est souvent très médiocre.*

La difficulté ne vient donc pas tant de l'imprimante elle-même que de son ruban encreur. Et cela nous a conduits à envisager un petit bricolage pour obtenir ce précieux ruban noir. Nous avons bravement rechargé la cartouche avec l'encreur qui sert aux tampons de la rédaction. Avec un peu d'habitude, c'est une opération rapide et qui n'est pas trop salissante... si l'on prend des gants ! Il faut recharger la cartouche de l'imprimante assez fréquemment, mais le résultat nous paraît prometteur.

Cela étant dit, ne suivez surtout pas notre exemple sans avoir mûrement réfléchi à l'avertissement qui figure dans la notice d'utilisation de la CE-122 (page 78) : « Attention : l'adjonction d'encre sur une cartouche de ruban encreur usée peut causer un dommage sérieux à l'imprimante ». Nous n'avons encore rien remarqué de catastrophique, mais nous n'avons pas suffisamment de recul : prudence donc.

Nous espérons parvenir rapidement à un résultat satisfaisant.

Pour la TI 57 (qui ne possède pas d'imprimante), nous avons résolu le problème des listes d'une tout autre façon. C'est un ordinateur individuel doté du programme approprié qui « se prend » pour une TI 57 et qui liste directement sur une imprimante à marguerite le programme à publier. Nous aurions pu faire de même pour le PC 1211 et le TRS 80 Pocket, mais le programme requis aurait été si long à écrire que nous aurions tout simplement manqué de temps pour réaliser l'Ordinateur de poche *et que les erreurs de recopie auraient pu être trop nombreuses.*

Et puis, puisqu'elle existe, il faut bien que la petite imprimante de Sharp serve à quelque chose.

Dernière minute : il semblerait que Tandy commercialise pour son imprimante des rubans noirs. Si c'est exact, et si (soyons prudents) ces rubans s'adaptent à l'imprimante Sharp, nous n'aurons plus besoin de nous salir les doigts !

A vos claviers

Le poquette aurait-il changé ?

Mon Sharp PC 1211 (acheté en juillet de l'année en cours) s'est montré, malgré toute ma bonne volonté, totalement indifférent aux manipulations suggérées dans l'article de la page 63 de votre premier numéro. Il m'est impossible d'accéder à ce compteur « hexadécimal » commandé par la touche ↧ . La même manipulation sur un TRS 80 Pocket moins récent (décembre 1980) fonctionne parfaitement.

Je serais curieux de connaître la raison de ce comportement « anormal » (sic). D'autres possesseurs de cette petite merveille ont-ils eu les mêmes difficultés ?

Marc Ferrer
69 Meyzieu

■ *Toutes les lettres que vous nous avez adressées depuis peu à ce sujet et ce que nous ont appris de vive voix diverses personnes qui ont acheté* ***dernièrement*** *un TRS 80 Pocket ou un Sharp PC 1211 concordent : il semble bien que la machine ait été modifiée et que dans ses versions les plus récentes elle ne permette plus qu'on aille visiter sa « face cachée », tout au moins au moyen du « sésame ». Ce sésame, nous l'avons vérifié, est efficace aussi bien sur les PC 1211 que sur les TRS 80 première version.*

Le constructeur aura sans doute corrigé le logiciel interne de sa machine de façon à éliminer cette anomalie. La machine ne présente donc plus ce petit défaut de la cuirasse. Est-ce pour la rendre impeccable ou pour éliminer tout risque d'indiscrétion ? Nous n'en savons rien.

Toujours est-il que les machines de la première génération qui se laissent forcer par le sésame, même si c'est un péché de jeunesse, peuvent paraître aux yeux de leurs détenteurs avoir quelque chose de plus que les autres... □

Bogue corrigée est à moitié pardonnée

■ Dans l'article collectif sur les dessous de la TI 57, il vous était entre autres choses proposer de " donner des lettres à votre TI 57 et d'autres choses étranges ". Et en effet, cela a dû paraître étrange à beaucoup, car les explications étaient très laconiques (quelques-unes avaient même purement et simplement disparu). Cela n'a pas empêché certains d'entre vous d'obtenir sur leur affichage les lettres attendues et même de creuser l'idée...

Nous vous redonnons donc la façon de procéder avec cette fois-ci un luxe de détails.

Allumez la machine et tapez LRN, l'affichage indique 00 00, tapez 2nd Exc (00 38 0 à l'affichage), SST (01 00), 2nd Lbl 1 (02 00), 3 (03 00), R/S (04 00), RST (05 00) et enfin LRN. La machine quitte le mode de programmation. Vous pouvez vérifier, en faisant LRN et BST que le pas n°4 contient bien l'instruction RST : 04 71. Réappuyez sur LRN.

Faites maintenant RST (0 à l'affichage), puis R/S (3 à l'affichage) et R/S de nouveau : la machine affiche toujours 3. En faisant LRN, vous découvrez que le pas n°4 a changé : 04 71 0 et, sans quitter le mode programmation, vous tapez un chiffre compris entre 0 et 9. Prenons 2 par exemple. L'affichage indique maintenant 05 00, ce qui est normal, mais si vous allez regarder le pas n°4 au moyen d'une pression sur la touche BST, vous découvrez 04 10, le code 10 ne correspondant à aucune fonction officielle sur la TI 57.

Appuyons de nouveau sur LRN et lançons l'exécution de ce programme insolite en pressant RST, R/S (affichage de 3) puis R/S de nouveau et vous verrez apparaître la lettre A. Si nous avions introduit 3 au pas n°4 en remplacement de 71 0, c'est la lettre b qui serait apparue, etc.

Forts de ces explications, vous pouvez maintenant reprendre en main le n°2 de *l'Op* et donner enfin des lettres à votre TI 57.

■ A la page 34 du n°2, la liste du jeu du chemin n'était pas tout à fait complète : il manquait le signe " moins " (–) au pas 287, entre 6 (pas 286) et XEQ 12 (pas 288).

VOUS AVEZ DIT INFORMATIQUE ?

Ma voiture, ma chaîne, ma perceuse... mon ordinateur. L'informatique est toujours plus présente et pourtant nombreux sont ceux qui ne savent pas réellement ce qu'est un ordinateur individuel, comment il fonctionne, et quelles sont ses applications.
Ces ouvrages jalonnent les trois étapes du chemin qui les fera passer de "l'ignorance" à la connaissance du contexte économique et social (L'ordinateur individuel) des notions fondamentales (Visa pour l'informatique) et des éléments techniques de base (Mon ordinateur) en informatique individuelle.

L'ordinateur individuel
La nouvelle ère de l'informatique
Par Yves Leclerc
Ecrit par un journaliste canadien, L'ordinateur individuel est, à la fois, le roman de l'informatique de Pascal à nos jours, une réflexion sur les implications sociales de l'informatique et une présentation de l'informatique individuelle, de ses applications présentes et futures.
240 pages - 55,00 FF

Visa pour l'informatique
Par Jean-Michel Jégo
L'informatique c'est quoi au juste ?
"Visa pour l'informatique" expose clairement ce qu'est l'informatique et ce à quoi elle sert. L'auteur définit les fonctions et organes essentiels d'un ordinateur et décrit ce qu'est un programme, en ne faisant intervenir qu'un nombre restreint d'instructions. L'ouvrage est complété d'exercices et d'exemples. "Visa" est donc bien le livre d'initiation à l'informatique tant attendu.
96 pages - 45,00 FF

Mon ordinateur
Par Jean-Claude Barbance
Comme tout objet technique, l'ordinateur, même individuel, ne peut être abordé sans une compréhension minimale de son fonctionnement et une connaissance de ses constituants.
"Mon ordinateur" s'adresse aux non initiés pour leur apporter cette compréhension et ces connaissances, et les aider à choisir un équipement.
Très pratique, ce livre est complété par un mini-dictionnaire des 140 termes et abréviations les plus utilisés en informatique, et une étude des prix à l'automne 1981.
128 pages - 60,00 FF

OP 3

Envoyer ce bon accompagné de votre règlement a
EDITIONS DU P.S.I.
41-51, rue Jacquard BP 86
77400 Lagny-s/Marne
Tél. (6) 007.59.31.

BON DE COMMANDE

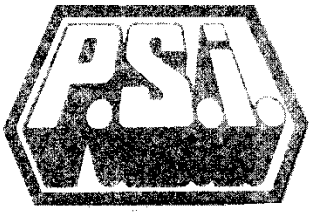
DESIGNATION	NOMBRE	PRIX
	TOTAL	

Les prix sont : taxes, emballage et port compris.
(par avion : ajouter 5 FF par livre)

NOM ..... PRENOM .....

rue ..... N° .....

Code post. |_|_|_|_|_| Ville .....



Editions du P.S.I.
41-51, rue Jacquard
BP 86
77400 Lagny-s/Marne
Téléphone (6) 007.59.31

au Canada : SCE Inc.
3449 rue Saint-Denis
Montréal Québec H2X3L1
Tél. : (514) 843.76.63

Magazine

Pour les mordus de la HP-41C

■ Ainsi que nous vous l'annonçions dans notre dernier numéro, les fanatiques (anglophones) de la programmation synthétique se régaleront à la lecture de l'ouvrage de W.C. Wickes *Synthetic programming on the HP-41C,* livre dont vous trouverez la critique ailleurs dans ce magazine. Signalons pour être complet deux autres sources d'information sur ces méthodes de programmation, qui utilisent les " codes cachés " de la HP-41C.

Tout d'abord, pour ceux que l'américain technique entrecoupé de termes " colloquial " ne rebute pas, il semble difficle d'ignorer le club *PPC* (Personal Programmable Calculator), ainsi que sa lettre d'information *PPC Journal.* Le club regroupe environ 3600 membres dans 46 pays et se consacre uniquement " au choix, aux soins, à l'utilisation et aux applications des calculateurs HP ", mais il est indépendant du constructeur. Le *Journal,* un mensuel, a animé aux Etats-Unis la découverte de la programmation synthétique et signale de nombreux " dessous " de la 41.

Les adhérents de PPC ont également découvert comment programmer directement en langage machine, c'est-à-dire sans passer par les instructions " de haut niveau ", *le microprocesseur de la 41C.* (Toutefois, il reste semble-t-il quelques micro-instructions à décoder : chercheurs, à vos machines !) L'utilisation de ces micro-instructions permet à la limite de doter la HP-41C d'un jeu d'instructions différent de celui fourni par HP : peut-être bientôt une TI-59 simulée sur 41, voire un langage BASIC ou Forth ?

Le club a même commandé à Hewlett-Packard un module de bibliothèque en mémoire MEM (la *PPC ROM*), qui doit être construit à 5000 exemplaires et contient une cinquantaine de sous-programmes et fonctions utilitaires destinés à la programmation synthétique et à la programmation en langage machine. *L'Ordinateur de poche* publiera un essai de cette bibliothèque dès que nous aurons réussi à nous en procurer un exemplaire, ce qui n'est en fait pas très facile : la plus grande partie des modules est réservée à des souscripteurs de la première heure.

La cotisation annuelle à PPC, avec l'abonnement " par avion " à la revue PPC Journal, coûte en Europe 31$ (environ 180 FF).

PPC Journal
c/o Richard J. Nelson
2541 W. Camden Place
Santa Ana, CA 92704
Etats-Unis

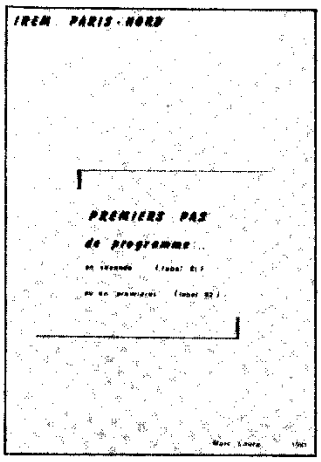
Une autre source d'informations, en français celle-là, est le mensuel *L'Ordinateur Individuel* (ce sont des amis, mais ce n'est pas une raison pour ne pas en parler !). Cette revue a notamment publié une série d'articles sur la programmation synthétique écrits par Philippe Descamps, Patrick Imbault, Bruno Langlois et Michel Sladki.

Ces articles parus dans les numéros 24 à 28 (février à juin 81) relèvent d'une démarche assez différente de celle de " l'école américaine " de PPC et Wickes, notamment parce que les découvertes accidentelles des fonctions cachées, si elles ont eu lieu à peu près à la même époque (début 80), ont été faites à partir de " fausses manipulations " différentes. L'ensemble des outils proposés par " l'école française " aboutit toutefois à des fonctions très semblables : les lecteurs français ne sont donc pas défavorisés, loin de là.

L'Ordinateur Individuel
41 rue de la Grange aux Belles
75483 PARIS CEDEX 10

Attention : la programmation synthétique amène parfois à des " plantages " mystérieux de la 41C ou CV, heureusement très rarement insurmontables ; nous ne connaissons aucun cas en fait définitif, mais il est conseillé de ne l'expérimenter qu'en connaissance des risques éventuels et des remèdes tout-puissants. Le constructeur de la machine se refuse à toute explication ou indication sur les " fonctions cachées ", et le seul recours en cas de panne réfractaire (rare rappellons-le) paraît être le retour de la machine en service après-vente. Il ne semble pas y avoir perte de la garantie, mais après cette période de garantie la facture risque d'être plus difficile à avaler ! □

■ UN LIVRE



Premiers pas de programme... en seconde ou en premières.
Marc Laura.
Université Paris-Nord
IREM, Villetaneuse
1981.
87 pages,
dactylographié.
Prix : 10 FF ttc.

■ Vous êtes enseignant, par exemple dans le secondaire, vous devez faire un cours de programmation avec des TI 57, vous n'avez pas la moindre idée d'un cours complet ? L'auteur veut apporter une réponse à votre problème.

Ce cours de 12 séances de 2 heures a été mis au point dans une classe de seconde. Il prend des élèves (et un prof !) complètement ignorants de la programmation et des calculatrices programmables, et les amène par des exercices de manipulation, de mise au point, de programmation, et même de construction graphique, à une maîtrise correcte de l'utilisation de la TI 57.

J'ai bien aimé le style simple et didactique de

Magazine

cette " leçon pour prof ". Prenons l'exemple de la première leçon. Elle comporte l'entrée d'un programme, sa mise au point et sa correction, puis un test. Le programme calcule un polynôme, on demande à l'élève de le calculer " à la main ", éventuellement avec l'aide de sa machine utilisée comme une " 4 opérations ", puis l'élève rentre le résultat et celui-ci est comparé au nombre trouvé par le programme. Cet exercice tout simple permet de plus de montrer astucieusement les problèmes liés à la précision limitée des calculs, et au fait que les calculs ne sont donc pas toujours rigoureusement exacts.

Des textes d'interrogations écrites, de devoirs, sont accompagnés de leurs corrigés assez largement commentés, ainsi que des résultats statistiques obtenus lors de l'expérimentation du cours : " tant d'élèves ont trouvé la méthode 1, tant la méthode 2, tant n'ont rien trouvé ".

A la fin du texte, l'auteur rapporte les réactions des élèves et leurs commentaires sur l'expérience, ainsi que les réponses qu'il fait à un certain nombre de leurs questions, réponses qui sont de fait *ses recommandations* pour qu'une expérience de ce type fonctionne correctement.

Ce petit ouvrage est sans prétention, y compris dans sa présentation ; quelques fautes de frappe et une impression parfois un peu pâlotte agaceront peut-être le lecteur, mais au vu de la modicité du prix, il est " difficile " d'être très difficile ! Je pense que cet ouvrage plaira à de nombreux enseignants, mais aussi à des débutants en programmation ayant tout de même un minimum de connaissances mathématiques (polynômes, fonctions rationnelles). Il faut sans doute commander directement cet ouvrage à l'IREM de Paris-Nord car il n'a semble-t-il été tiré qu'à très peu d'exemplaires :

IREM Paris-Nord
Avenue Jean-Baptiste Clément
93430 VILLETANEUSE

□ BS

Surprise pour les fanatiques de Texas Instruments

■ La surprise, c'est hélas qu'il n'y a pas de surprise : vous devez vous sentir bien seuls en ce moment, TI semblant toujours dans une expectative assez complète quant à de nouveaux modèles. Devrait cependant être introduite très prochainement une TI-51 avec affichage à cristaux liquides, fonctions scientifiques et une trentaine de pas de programmes, aux environs de 300 FF ttc : ce n'est pas une révolution, mais peut-être un test de l'affichage à cristaux liquides. A quand la suite ? Il n'y a en tout cas à notre connaissance aucune revue, aucun club similaires à ceux dont peuvent de leur côté disposer les " adeptes de la religion polonaise ".

□

Du côté des ordinateurs

Micropoche ou pas ?

■ Ainsi que nous le signalions dans notre premier numéro, le **Sinclair ZX-81** ni son prédécesseur le ZX-80 ne sont des ordinateurs de poche, puisqu'ils doivent être connectés à un écran de télévision. Mais après tout le ZX-81 coûte moins que certains ordinateurs de poche à langage BASIC ou LMS (Langage Machine Spécialisé), et avec une petite TV son volume n'est finalement guère plus gros qu'un actuel micropoche complet avec tous ses accessoires. Et comme Sinclair a introduit récemment une TV " de poche " à écran plat, dont le volume de fabrication est prévu (avec optimisme ?) en 1982 à un million d'unités... Quoi qu'il en soit, le ZX de l'année est maintenant disponible en France, avec un clavier et surtout un BASIC et un affichage TV notablement améliorés par rapport à celui de l'année dernière. On regrettera par contre qu'il n'y ait toujours pas de mémoire permanente (le stockage sur cassettes semblant parfois poser des problèmes), et que le constructeur ne prévoie pas, en plus de la connexion à un écran TV extérieur, un affichage d'une ligne à cristaux liquides : la comparaison avec un PC 1211 ou d'un Matsushita lui serait favorable en tous points, puisque son prix plus bas serait associé à la portabilité qui est (par définition) l'un des points forts de ces vrais micropoches. En tout cas, comme cela se produit souvent pour les calculatrices, le nouveau modèle est plus puissant et nettement moins cher que le modèle précédent (985 FF ttc au lieu de 1250, soit 21% de moins). Une imprimante optionnelle peut se connecter sur le ZX-81 (ou 80, moyennant un changement de circuit), au prix de 690 FF ttc.

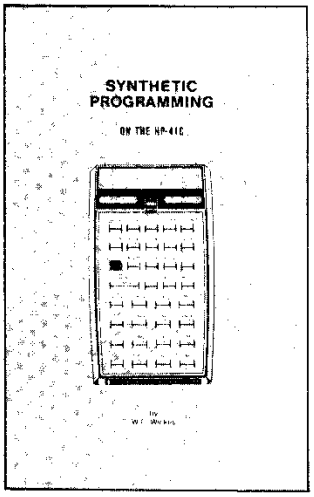
L'importateur du Sinclair, la société **Direco International** (téléphone : (1) 359.72.50), affirme avoir vendu en 6 mois plus de 7 000 ZX-80 en France : un tel rythme placerait bientôt dans notre pays les ZX en tête du classement du parc d'ordinateurs " causant BASIC ".

Il ne reste plus qu'à attendre (on peut rêver) la cuvée 82 du ZX, que l'on pourrait par exemple appeler ZX-82, et qui comporterait en un seul boîtier une unité centrale à mémoire permanente associée à un écran uniligne à cristaux liquides, le tout fonctionnant sur de petites piles, et, toujours dans le même boîtier mais avec une source d'alimentation extérieure, un écran de TV extra-plat ; autrement dit, un véritable " poquette " qui serait commercialisé en France à 700 FF ttc ?

□

Magazine

■ UN LIVRE



Synthetic programming on the HP 41C.
William C. Wickes
Larken Publications, College Park MD, 1981 (4e édition).
92 pages, dactylographié.
Prix : 149 FF ttc.

■ Cet ouvrage est une synthèse des différents articles parus dans le *PPC Journal*, que nous présentons par ailleurs, sur la programmation " synthétique " des systèmes HP 41C et CV. A partir des découvertes signalées par de nombreux membres de PPC, dont Wickes lui-même, et publiées dans le *Journal*, il était en effet nécessaire de réunir les différentes informations obtenues et d'en faire un tout cohérent, afin que " les instructions cachées " de la 41 puissent être un véritable outil et non plus un simple objet de curiosité.

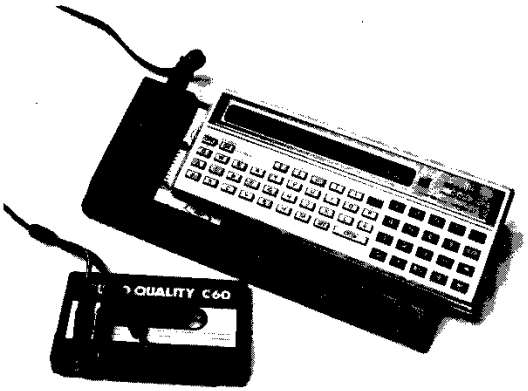
La programmation synthétique, c'est tout simplement la synthèse d'instructions de la HP 41C à partir de morceaux divers : autres instructions, caractères alphanumériques, etc. Il s'avère que l'on peut ainsi créer en particulier des instructions qui officiellement n'existent pas (puisque HP ne les mentionne ni ne les décrit dans sa documentation, et se contente de dire " no comment " à toute question sur le sujet), mais qui ont une existence bien réelle puisqu'on peut les créer et les exécuter. L'avantage, c'est que ces fonctions sont très puissantes : elles permettent notamment d'accéder de façon " simple " à certains registres internes. L'inconvénient, c'est que cette puissance est parfois dangereuse, justement parce que certains registres sont " sensibles ", et qu'y stocker une valeur inadéquate peut suffire à " planter " la machine.

Ce sont toutes ces manœuvres bizarres, et les résultats qu'elles permettent d'obtenir, qui sont décrites dans *Synthetic Programming*. Au passage, le lecteur est obligé bon gré mal gré de se plonger dans la structure interne de la 41 et le fonctionnement de ses différentes instructions, " officielles " ou non. Le texte est complété par de nombreuses illustrations, tables et listes de programmes dont l'apport est très appréciable. Certes, les programmes malgré leur intérêt sont parfois un peu lassants à rentrer... mais comme ils sont également donnés en code barre à la fin du livre, même les paresseux ou les distraits pourront les utiliser.

Ce livre est bien entendu formellement déconseillé à tout autre qu'un fanatique de la HP 41C doté d'un bagage suffisant d'anglais. Il ne sera sans doute pas inutile, avant d'aborder la lecture de ce bouqin, de jeter un œil aux articles sur le même thème publiés dans *L'ordinateur Individuel* (quand on vous dit ce sont des amis...) par des auteurs " de l'école française " qui suivent une démarche différente de celle de " l'école américaine " de Wickes. On pourra par contre s'interroger sur le prix de *Synthetic Programming* en France : 1,60 FF ttc par page dactylographiée, n'est-ce pas un peu élevé ?

□ BS

ORDINATEUR SHARP PC 1211



DISPONIBLE AVEC TOUS LES MANUELS D'APPLICATIONS

PRIX GARANTIS JUSQU'AU 15-10-1981

ORDINATEUR PC 1211	1 250 F TTC
INTERFACE IMPRIMANTE	1 000 F TTC
INTERFACE MAGNETOPHONE	150 F TTC

CREDIT GRATUIT 3 MOIS

MICROEDITIONS INFORMATIQUES
116, RUE DE PARIS - 93100 - MONTREUIL
TEL. : 857.96.33 - METRO ROBESPIERRE

Chauffez-vous avec un micropoche

■ La société **Etude Solaire** (téléphone (1) 569.63.73) propose pour les HP 41C et CV des programmes destinés aux chauffagistes. Ces derniers peuvent les utiliser pour résoudre divers problèmes de calcul liés aux installations de chauffage traditionnel, basse température ou énergie solaire.

Parmi les divers programmes disponibles : calcul heure par heure de la position du soleil et des énergies directe, diffuse, globale et transmise sur 1 m² de vitrage en fonction de la position, de la latitude et de la déclinaison solaire (programme RST/RSH, 710 FF ttc) ; calcul de l'énergie solaire utile récupérée par 1 m² de capteur en fonction de ses caractéristiques, de la déclinaison solaire, de la fraction insolation moyenne mensuelle, et de l'écart de température entre le fluide du capteur et l'extérieur (programme ESU, 470 FF ttc) ; calcul du coefficient G d'un logement en fonction du DTU règles THK 77 et THG 77 (programme C.G., prix 1 650 FF ttc). □

■ UN LIVRE

La découverte du PC 1211.
Jean-Pierre Richard.
Editions du PSI,
Lagny, 1981.
Broché 152 pages.
Prix : 75 FF ttc.

■ Pour profiter vraiment de ce livre, il vous faudra avoir sous la main l'ordinateur (Sharp ou TRS), bref l'OP pour lequel il a été écrit. Que va découvrir le lecteur ? Beaucoup. Et sa machine entre autres choses...

A la lecture, tout se passe en effet comme si vous aviez avec vous quelqu'un d'aimable qui vous guide en vous montrant ce qu'il faut faire et ce qu'il faut éviter. On commence tout " bêtement " par le début, et l'on progresse petit à petit, exemple après exemple.

C'est ainsi qu'on pourra, par la pratique, s'accoutumer au maniement d'un petit ordinateur BASIC. Tout compte fait, ce livre semble assez bienvenu. On retiendra surtout qu'il a été conçu pour ceux qui utilisent un PC 1211 ou un TRS 80 Pocket, c'est-à-dire pour des personnes dont la plupart découvrent le BASIC avec leur machine. A cet égard, ceux et celles qui connaissent déjà la programmation trouveront que ce livre contient beaucoup d'évidences, mais les autres seront " initiés " en douceur. □ JPB

Magazine

Les poquettes japonais prolifèrent

■ Quelques nouveautés en cette rentrée. Un timide essai donc chez **Texas** avec une programmable de faible capacité à cristaux liquides. Deux nouvelles venues de **Hewlett-Packard**, les HP-11C et 12C qui ne sont d'ailleurs pas non plus révolutionnaires. Vous pouvez lire un essai de la 11C dans le présent numéro.

L'ordinateur de poche de **Sanyo** se fait toujours attendre, mais le **Matsushita/Quasar** est enfin arrivé, même si sa commercialisation n'a pas encore commencé : vous pouvez également en lire un essai dans le présent numéro.

Depuis notre dernier numéro, **Casio a annoncé deux nouvelles machines, la FX602-P** et la **FX702-P**. Il ne semble pas pour l'instant que la 602 soit appelée à être commercialisée prochainement en France ; d'après l'un de nos correspondants alsaciens, il s'agit d'une calculatrice scientifique programmable actuellement disponible en Allemagne. Elle dispose d'un affichage à cristaux liquides et de la mémoire permanente, avec une capacité de 512 pas de programme, de 22 mémoires et de 10 niveaux de parenthèses, ainsi que de possibilités alphanumériques, bref une réponse à la 41C de HP. Le prix actuel en Allemagne est de 300 DM (environ 720 FF), et de 80 DM (200 FF) pour la nouvelle interface cassette FA-2. Une version simplifiée, la **FX601-P** coûte 230 DM (550 FF) sans interface, et possède une capacité de 128 pas, 11 mémoires et 5 niveaux de parenthèses : ce modèle est sans doute appelé à terme à remplacer la 502 qui semble, tout au moins en France, n'avoir jamais rencontré le succès qu'aurait pu lui valoir ses qualités.

En ce qui concerne la 702, il s'agit d'un véritable ordinateur de poche programmable en BASIC d'une capacité assez proche du PC 1211, mais beaucoup plus performant : jusqu'à dix fois plus rapide, plusieurs tableaux de variables, tableaux à deux dimensions et fonctions de chaînes de caractères moins rudimentaires (mais encore loin de la perfection). Le clavier comporte 65 touches *placées dans l'ordre alphabétique,* et il est surmonté d'un afficheur à cristaux liquides de 20 caractères dans une matrice 5 × 7 avec (!) possibilité de réglage de contraste de l'affichage. Ce micropoche peut être complété au moyen de la nouvelle interface cassette FA 2 ainsi que par la future imprimante FP 10 ; elles s'adaptent toutes deux également aux 602, 601 et 502. Les prix aux Etats-Unis ont été initialement annoncés comme étant de 200 $ (1 200 FF environ) pour le 702 et 80 $ (460 FF) pour l'imprimante optionnelle, et au Japon 40 000 et 16 500 yens respectivement (environ 1 050 et 420 FF). Il serait également question, à un prix non précisé, d'une extension optionnelle de la mémoire vive MEV, ce qui augmenterait notablement la capacité de la machine en pas de programmes. Vous en trouverez un essai complet, réalisé par notre correspondant japonais, dans notre prochain numéro (en vente mi-décembre, si, si !). *Dernière minute : la commercialisation en France de la 702 devrait commencer en octobre, à un prix avoisinant 1 300 FF ttc. Ceci laisse espérer pour l'imprimante un prix de l'ordre de 1 300*(80/200) = 520 FF ttc environ : correct.*

Peut-être est-ce à cause du Casio qu'aux Etats-Unis, **Sharp** a baissé les prix du PC 1211 et de ses accessoires. Le PC passe de 250 à 180 $ (1 400 à 1 000 FF), une baisse de 25% *qui n'est pas due* à la variation du cours du dollar, puisque celui-ci est resté assez stable par rapport au Yen japonais. L'interface imprimante passe de la même façon de 150 à 130 $ (860 à 750 FF). Remarquons que, s'il est peu probable que les prix européens du PC 1211 varient (maintenir les prix actuels revient à " amortir " l'augmentation du cours du dollar par rapport aux monnaies européennes), le prix de l'imprimante semble toujours aussi exagéré puisque très voisin de celui du " poquette ", alors que ces prix sont nettement différents aux Etats-Unis. (Nous avons cependant aperçu l'imprimante à 895 FF ttc chez Tandy et 850 FF ttc chez JCR). Peut-être Sharp *et Tandy* pourraient-ils faire un effort de ce côté ? **Tandy**, en tout cas, a purement et simplement augmenté en Europe les prix du PC 1211 (maintenant à 1 395 FF ttc), alors même qu'il semble probable qu'aux Etats-Unis les prix seront alignés sur ceux de Sharp, voire inférieurs... □

On a souvent besoin d'un petit micropoche pour prendre une décision, même si l'on travaille dans une grande société

Des ordinateurs de poche pour...

L'Ordinateur de poche : quelle est exactement votre fonction ?

Gérard D. : je suis responsable de " l'outil de distribution " d'un important groupe alimentaire. En tant que tel, je m'occupe notamment de tout ce qui concerne la logistique de nos produits, depuis l'instant où ils arrivent dans nos dépôts jusqu'au moment où ils sont payés — bien entendu après avoir été préparés, facturés et livrés.

Pour la seule distribution des produits, j'interviens non seulement pour choisir le nombre, l'implantation et l'organisation des dépôts, mais aussi pour organiser les tournées de prise de commande et de livraisons auprès des clients. Une partie de notre production est conditionnée sous forme de conserves, mais pour plus de 80 %, il s'agit de produits frais ou en sachets : principalement de la charcuterie.

Nous sommes donc tenus de maintenir une chaîne de froid continue : camions isothermes, dépôts à température régulée entre 0 et 5°C. Nous devons de plus livrer nos clients dans des délais suffisamment brefs car nous apposons au moment de la fabrication une date limite de vente ; de ce côté-là, nous sommes soumis — très normalement d'ailleurs — à des contraintes réglementaires plus dures que celles qui s'appliquent, par exemple, aux produits laitiers.

Nos clients sont tout aussi bien des hypermarchés et des supermarchés que de tout petits commerçants ou des collectivités : il n'y a pratiquement pas de grossistes dans notre profession. En ce qui nous concerne, nous vendons tous nos produits livrés chez les clients.

Quelle est l'importance économique de la livraison par rapport aux autres fonctions de la distribution ?

■ Elle est loin d'être négligeable. Comme tout le monde, nous essayons bien sûr de diminuer nos coûts. Jusqu'à une date récente, les coûts de livraison n'étaient pas la préoccupation majeure des responsables de distribution. La gestion des stocks, la facturation et l'organisation du flux des produits étaient privilégiées. Disons tout de suite que si notre attention a été attirée sur ce point, les hausses du prix du pétrole n'y sont pas pour rien ! Mais il y a une autre raison elle aussi très importante : nous voulons que nos chauffeurs-livreurs aient des horaires " normaux ", et nous y veillons avec soin. La législation à d'ailleurs prévu toute une série de dispositions (amplitude horaire, repos compensateurs par exemple) qui vont dans ce sens. Cet aspect humain du problème n'est pas du tout secondaire.

Quels sont les éléments qui rentrent en ligne de compte dans le choix de l'implantation des dépôts ?

■ Le choix de l'implantation des dépôts s'est fait et continue à se faire en fonction de deux contraintes contradictoires. Pour résumer, je dirai que l'on a intérêt d'une part à ce que le rayon d'action autour d'un dépôt soit aussi petit que possible : on gagne alors sur les temps de livraison. Mais d'un autre côté, on a intérêt à avoir des dépôts aussi grands que possible, puisque cela permet de minimiser les stocks immobilisés et les coûts fixes. L'optimum économique se trouve donc quelque part entre ces deux extrêmes. Actuellement, la plupart de nos dépôts effectuent de 250 à 800 livraisons par jour en moyenne.

Approvisionner un petit dépôt éloigné mais dont les tournées de livraisons seront courtes ou un grand dépôt central plus proche qui fera des tournées plus longues, cela ne revient-il pas à peu près au même ?

■ Non, car la longueur des tournées augmente le temps de conduite et diminue le nombre de livraisons. Il a fallu trouver un compromis ; c'est ce que j'appelle les centres d'éclatement. Cela consiste à faire faire à un gros camion une partie de la distance et à répartir son contenu entre plusieurs camionnettes basées au-delà du rayon d'action de 80 à 90 km que l'on considère comme normal. Le transport d'approche s'effectue de nuit, ce qui laisse aux camionnettes toute la journée pour livrer.

Pour introduire une image sur laquelle je reviendrai sans doute par la suite, je distinguerai deux types de parcours que peut effectuer un camion : il y a d'abord le " cabotage " qui correspond aux livraisons à courte distance autour d'un centre géographique. Dans un rayon donné, le camion passe d'un point de livraison à l'autre : les distances entre deux arrêts sont courtes, et les arrêts aussi nombreux que possible.

Le deuxième type de parcours s'apparente plus à une navigation " au long cours " : le véhicule plein ou vide se rend d'un point à un autre sur une distance qui peut être grande. Lorsque le camion se rend du dépôt au centre géographique à partir duquel vont se faire les livraisons, nous parlons " d'approche ".

Cette distinction entre l'approche et le cabotage est en fait essentielle à la plupart des problèmes de distribution tels que nous les concevons : il ne faut pas considérer de la même façon les kilomètres effectués pendant l'approche et ceux qui sont effectués pendant le cabotage.

Pour prendre une autre comparaison, j'évoquerai l'abéille. De la ruche au massif de fleurs, elle parcourt une certaine distance (disons cinquante mètres), c'est une chose. Puis, en butinant dans le massif, elle effectue un grand nombre de petits trajets : elle fait du cabotage. Si elle veut optimiser sa récolte du pollen, je pense qu'elle ne doit pas confondre l'aller-retour ruche-massif et la distance parcourue de fleur en fleur. Plus la distance de la ruche au massif sera grande, moins l'abeille aura de temps pour butiner dans sa journée et donc moins elle aura récolté de pollen.

Pour vos études, quels moyens de calcul utilisez-vous ?

■ Cela va de très petites machines (des ordinateurs de poche HP 41C) au très gros ordinateur IBM central de la société, sans oublier un ordinateur " de table ", un HP 9831.

Mais vous-même, qu'utilisez-vous ?

■ Uniquement ma HP personnelle que je mets un peu à toutes les sauces, non seulement pour mon travail, mais aussi dans le cadre de mes autres activités.

Existe-t-il des algorithmes qui permettent de résoudre votre problème ?

■ Oui, il en existe, mais justement ils ne sont pas adaptés à notre cas. L'algorithme classique " en pétales de marguerite " considère que toutes les distances, approche ou cabotage, sont équivalentes : il ne cherche pas à rentabiliser la phase d'approche, c'est-à-dire à l'amortir sur le plus grand nombre possible d'opérations de cabotage.

Or l'un des éléments importants dans le coût de nos livraisons, c'est ce que nous appelons le prix du " coup de frein ". Le prix du " coup de frein " correspond aux coûts fixes (véhicule plus chauffeur) divisés par le nombre de livraisons. L'algorithme classique " interdistance " ne tient pas tellement compte de ce prix du coût de frein, et c'est principalement pour cela que nous ne l'utilisons pas. Mais il y a d'autres raisons.

Sur un plan purement théorique, c'est une très jolie méthode, mais dans la pratique on s'aperçoit que ses résultats ne peuvent jamais être vraiment mis en application. De plus, ces algorithmes, du type " Branch and Bound " sont extrêmement gourmands en capacité mémoire et en temps de calcul. Et dans une grande entreprise, il n'est guère raisonnable de dépenser des fortunes en temps de calcul pour obtenir un résultat optimum sur le plan théorique mais inapplicable en grande partie. Il faut donc adopter une attitude plus pragmatique.

Alors que faites-vous ?

■ Nous utilisons un algorithme beaucoup plus " réaliste " qui nous donne non pas le meilleur résultat théorique, mais un bon résultat pratique. En se fixant ce but, on s'autorise un certain nombre de simplifications avec, dans tous les cas, une réduction énorme du temps de calcul.

C'est ainsi qu'avec l'algorithme mis au point par un ingénieur-conseil extérieur, Gérard H., il devient possible d'obtenir rapidement sur un ordinateur de poche un résultat très correct. Nous sommes limités, puisque nous ne pouvons traiter que des problèmes portant sur quelques centaines de points de livraison, mais le plus souvent c'est amplement suffisant, car nous pouvons notamment découper les problèmes. Et pour les cas plus complexes nous utilisons l'ordinateur de table dont nous disposons : la limite passe alors à près de 2 000 points.

Les résultats donnés par la " grosse " machine peuvent d'ailleurs être ensuite découpés en secteurs suffisamment petits pour être traités sur l'ordinateur de poche. Le gros avantage, c'est qu'il devient possible d'aller optimiser " sur le terrain " : à partir des indications fournies par l'ordinateur de table, nous pouvons découper l'aire d'activité d'un dépôt en plusieurs secteurs et sauver sur fiches magnétiques les données correspondant à chaque secteur. Nous définissons ainsi des cartes et des secteurs et c'est au dépôt que nous constituons les tournées en tenant compte le cas échéant d'autres nécessités, ou en envisageant d'autres découpages. C'est donc avec l'aide du personnel du dépôt que nous affinons la solution, et c'est important : certains détails que nous ne pouvons pas connaître depuis Paris pèsent lourd.

Comment votre algorithme se débrouille-t-il avec des contraintes comme les points de passage obligé ou l'existence d'une route de montagne ?

■ Vous voulez aborder les cas particuliers. C'est exact, une route de montagne fait en quelque sorte exception puisque l'itinéraire se trouve pratiquement imposé et que la distinction entre cabotage et approche s'estompe.

Il y a d'ailleurs d'autres cas particuliers tels que le franchissement d'un fleuve ou l'entrée sur une autoroute qui ne peuvent se faire qu'en un nombre limité de points. Il y a même pire : les culs-de-sac, c'est-à-dire les trajets avec point de passage obligé à l'aller comme au retour. Eh bien, pour ces points de passage obligé, notre algorithme utilise des changements d'origine. En gros, cela revient à considérer que l'ensemble des livraisons ayant ce point de passage obligé sont effectuées à partir de lui. On étudie alors une ou plusieurs tournées à partir de ce point.

Comment votre opération est-elle mise en œuvre dans les dépôts ?

■ Comme je l'ai dit précédemment, nous définis-

Pour la plupart, ce sont de petits programmes qui m'assistent dans mon travail

sons des cartes et des secteurs. Dans un deuxième temps, moi-même ou une personne de mon service nous nous rendons dans le dépôt avec une HP 41C, généralement deux fois par an. Et c'est sur place que sont déterminées les " zones de cabotage " des tournées avec les corrections manuelles qui s'imposent parfois.

Pourquoi deux fois l'an ? Parce que beaucoup de nos clients ont une activité saisonnière avec pointes en été ou en hiver selon les régions. Nous calculons donc les tournées et pour la période de saison et pour la période hors-saison.

Et comment les dépôts acceptent-ils votre intervention ?

■ Plutôt bien, parce que nous n'imposons jamais une solution : nous proposons de bonnes lignes d'orientation, sans obligation.

Quels résultats espérez-vous obtenir grâce à cette opération ?

■ Il est difficile de donner un chiffre. Disons qu'en diminuant le nombre des tournées trop peu remplies — l'essentiel du gain se trouve là, — je pense que nous allons économiser 10 à 15 % de nos frais de livraison en valeur absolue.

Vous avez dit tout à l'heure que vous utilisiez votre micropoche " à toutes les sauces ". Pouvez-vous donner d'autres exemples ?

■ Bien sûr. Comme dans toutes les entreprises, nous utilisons des tableaux de bord. Dans mon cas, il s'agit d'un suivi du chiffre d'affaires : le contrôle d'activité. Cela n'a rien de bien sorcier. Il suffit de comparer les résultats du mois passé avec ceux du même mois de l'année précédente. On effectue des cumuls sur l'année et l'on obtient des pourcentages d'écart ou de variation. Bref, il s'agit surtout d'additions, de soustractions et de pourcentages. Bien entendu, il serait particulièrement peu économique de confier ces opérations à un gros ordinateur, sans compter que la saisie des informations prendrait plus de temps que le calcul à la main !

Voilà pourquoi, chaque mois, pendant deux jours et demi, une personne faisait à la main il y a quelques années les calculs correspondants. Nous avons ensuite utilisé une calculatrice assez simple, et le même travail s'est fait en une demi-journée. Maintenant, grâce aux cartes magnétiques de mon micropoche, cela me prend très exactement une demi-heure. Je conserve en effet d'un mois sur l'autre les données sur cartes magnétiques. Il me suffit donc de rentrer les nouvelles données et de les sauver sur cartes : le programme se charge de me fournir tous mes résultats imprimés.

Cela ne vous fait-il pas perdre du temps ?

■ Ça me prend du temps, mais je ne dirais pas que c'est du temps perdu. En travaillant sur les chiffres, je suis amené à " piocher " un peu plus certains résultats. Et comme j'ai sous la main un moyen de calcul pour vérifier diverses hypothèses, je ne m'en prive pas !

En fait, je garde constamment ma machine avec moi et j'ai toujours un petit programme en cours.

Quel genre de programmes avez-vous écrits ?

■ Pour la plupart, ce sont de petits programmes qui m'assistent dans mon travail. Mais aussi des programmes plus importants de calcul et d'impression de répartition de dépenses, un autre pour le calcul des primes salariales, d'autres encore pour le calcul des impôts, pour l'établissement des coefficients saisonniers, etc. D'autre part, dans mes activités de maire-adjoint, mon micropoche me sert également pour les calculs financiers habituels : mensualités et amortissement d'un prêt, dettes de la commune, tableaux financiers, etc. J'utilise même un programme pour le calcul des impôts locaux qui est assez similaire à celui décrit dans votre dernier numéro. Naturellement, c'est moi qui l'ai écrit, et je trouve mon programme meilleur !... De toute façon, je ne pouvais pas faire autrement car je suis imperméable à la notation algébrique utilisée par l'autre programme. Converti de longue date à la religion de la notation polonaise, il fallait que j'écrive mon propre programme.

Comment faites-vous pour vous retrouver au milieu de tous ces programmes ?

■ Tout au début, en 1979, quand je n'avais pas de lecteur de cartes ni d'imprimante, je recopiais soigneusement les listes de mes programmes dans un carnet. Maintenant, avec les cartes, c'est devenu inutile. Mais je continue tout de même à tout répertorier et je note pour moi le mode d'emploi de tous les programmes : je tiens à pouvoir les utiliser même après deux ou trois ans.

A mes débuts sur HP41C, j'avais d'ailleurs commencé par écrire dans mon carnet quelque chose qui fait défaut dans le manuel du constructeur et qui m'a fait gagner du temps : la liste manuelle de *toutes* les instructions avec pour chacune d'entre elles *les manœuvres nécessaires pour l'obtenir.* Certaines sont en effet assez complexes et elles n'ont vraiment rien d'évident quand on ne sait pas encore utiliser la machine.

☐ Propos recueillis par Jean-Baptiste Comiti

Avez-vous une mémoire aussi bonne que votre xxxpoche ?

000	76	LBL	047	45	Y^x	094	09	9
001	12	B	048	43	RCL	095	09	9
002	01	1	049	01	01	096	09	9
003	02	2	050	42	STO	097	09	9
004	42	STO	051	00	00	098	09	9
005	10	10	052	43	RCL	099	09	9
006	43	RCL	053	01	01	100	09	9
007	04	04	054	22	INV	101	09	9
008	42	STO	055	44	SUM	102	09	9
009	02	02	056	11	11	103	91	R/S
010	42	STO	057	25	CLR	104	43	RCL
011	03	03	058	76	LBL	105	01	01
012	76	LBL	059	35	1/X	106	42	STO
013	95	=	060	91	R/S	107	00	00
014	01	1	061	32	X:T	108	01	1
015	44	SUM	062	73	RC*	109	22	INV
016	10	10	063	11	11	110	44	SUM
017	44	SUM	064	22	INV	111	03	03
018	01	01	065	67	EQ	112	61	GTO
019	43	RCL	066	32	X:T	113	52	EE
020	10	10	067	01	1	114	76	LBL
021	42	STO	068	44	SUM	115	33	X²
022	11	11	069	11	11	116	36	PGM
023	43	RCL	070	97	DSZ	117	15	15
024	01	01	071	00	00	118	71	SBR
025	42	STO	072	35	1/X	119	88	DMS
026	00	00	073	43	RCL	120	65	×
027	71	SBR	074	01	01	121	01	1
028	33	X²	075	22	INV	122	00	0
029	72	ST*	076	44	SUM	123	00	0
030	11	11	077	11	11	124	95	=
031	76	LBL	078	97	DSZ	125	59	INT
032	52	EE	079	02	02	126	92	RTN
033	01	1	080	95	=	127	76	LBL
034	03	3	081	43	RCL	128	10	E'
035	42	STO	082	03	03	129	32	X:T
036	11	11	083	65	×	130	47	CMS
037	76	LBL	084	02	2	131	32	X:T
038	45	Y^x	085	00	0	132	36	PGM
039	73	RC*	086	55	÷	133	15	15
040	11	11	087	43	RCL	134	15	E
041	66	PAU	088	04	04	135	92	RTN
042	01	1	089	95	=	136	76	LBL
043	44	SUM	090	91	R/S	137	11	A
044	11	11	091	76	LBL	138	42	STO
045	97	DSZ	092	32	X:T	139	04	04
046	00	00	093	09	9	140	92	RTN

Combien de nombres de deux chiffres pouvez-vous retenir d'affilée ? Trois, cinq ou dix ? Voici un petit jeu pour TI 58 ou TI 59 qui vous le dira. Si vous le pratiquez assez longtemps, vous pourrez même mesurer vos progrès.

■ On peut classer les jeux qui se disputent à l'aide d'ordinateurs (de poche ou non) en deux catégories au moins : les jeux qui existaient avant l'informatique (échecs, bataille navale, othello, morpions, etc.) et ceux que l'informatique a permis de créer. C'est à cette dernière catégorie qu'appartient le jeu que nous vous proposons. Il s'agit en fait de la version numérique d'un jeu qui connaît beaucoup de succès sous sa forme sonore et visuelle (cf. le jeu *Simon,* entre autres).

Ce jeu fait essentiellement appel à la mémoire du joueur et constitue d'ailleurs un excellent exercice de ce point de vue. Et ce qui me plaît le plus, c'est qu'il est très simple, ce qui est à mon avis une belle qualité pour un jeu.

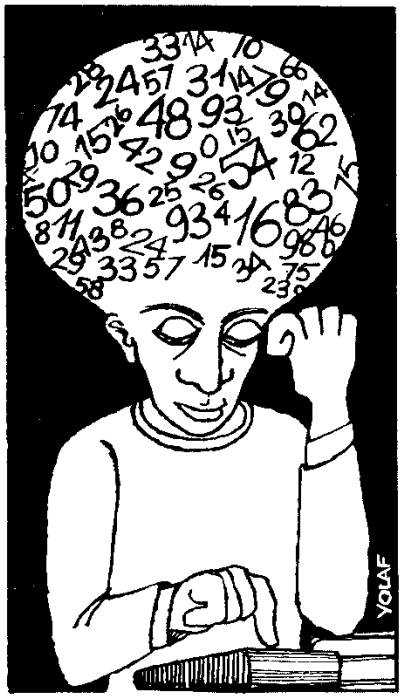
Le principe en est le suivant : l'xxxpoche (ici une TI 58 ou 59) affiche d'abord pendant un bref instant un nombre compris entre 0 et 99 et vous devez répéter ce nombre. Si vous ne vous êtes pas trompé, la machine vous répète ce premier nombre et en ajoute un autre. Vous devez réécrire le tout au clavier, et ainsi de suite jusqu'à ce que vous ayez un trou de mémoire. Mais je pense qu'il sera beaucoup plus parlant de vous commenter une partie.

Après avoir inscrit le programme dans la machine, vous affichez un nombre quelconque et vous

appuyez sur E'. Pour suivre exactement cette première partie, introduisez 18. Après E' (autrement dit 2nd E), le nombre que nous venons d'inscrire réapparaît à l'affichage : 18. Il faut alors indiquer au programme combien de nombres successifs nous pensons réussir à retenir. Dans notre exemple, ce sera 5. On inscrit donc 5 et l'on appuie sur la touche A. Il suffit maintenant d'appuyer sur B pour que le jeu débute.

Le temps d'une " pause ", un premier nombre apparaît : 70. Puis la machine affiche 0. A nous d'inscrire 70 et d'appuyer sur R/S. L'afficheur nous montre alors, de façon fugitive, 70 puis 21. Nous inscrivons donc 70 (R/S) et le 1 qui apparaît nous indique que notre réponse est bonne. Inscrivons 21 et enfonçons la touche R/S. Ce sont maintenant 3 nombres qui défilent : 70, 21 et 76.

Inscrivons ces trois nombres en les faisant suivre chacun par R/S. Nous voyons maintenant 4 nombres se succéder : 70, 21, 76 et 54. C'est de nouveau à nous de jouer, et nous commettons exprès (une fois n'est pas coutume) une erreur : nous inscrivons 71 à la place de 70 et nous



pressons sur R/S. Une brochette de 9 illumine l'affichage pour nous avertir que nous venons d'avoir une défaillance. Il suffit d'appuyer sur R/S pour que l'ordinateur de poche nous rappelle la suite précédente : 70, 21, 76 et 54, et nous donnons la bonne réponse 70 R/S 21 R/S 76 R/S 54 R/S. Une cinquième et dernière suite apparaît : 70/21/76/54/43.

Nous refaisons comme précédemment et cette fois-ci nous remarquons qu'après avoir entré 43 R/S, un nombre demeure à l'affichage : 16. C'est notre note sur 20 qui a été calculée en fonction des erreurs commises et du nombre d'essais demandés en A : nous en avions demandé 5 et nous en avons raté un.

Sans parler de la TI 59, sur la seule TI 58, en changeant la partition de la mémoire, on peut obtenir jusqu'à 30 nombres consécutifs, et je pense que c'est plus que suffisant, même pour les personnes qui ont une mémoire d'éléphant.

☐ Denis Alexandre

initiation

Un programme basic sous la loupe La vie des microbes

Jouer avec un micropoche est une des façons sérieuses (?) de progresser dans l'art de programmer.

■ C'est entre Paris et Rennes, dans une petite voiture qui m'emmenait en vacances que j'ai écrit ce programme. Il y avait ce jour-là de sérieux ralentissements, et tout compte fait cela m'aura pris beaucoup de temps. Inutile de préciser que je n'étais pas au volant.

Une fois arrivée à bon port (c'est le cas de le dire puisque notre destination finale était Douarnenez), il me restait à apporter quelques retouches de pure forme à cette première version un peu " brouillon ", en particulier à en renuméroter les lignes pour donner meilleure apparence à l'ensemble, et le tour était joué.

Vous remarquerez, je l'espère, l'effort auquel je me suis astreinte pour respecter scrupuleusement les règles du langage BASIC. Contrairement à mon habitude, j'ai soigneusement veillé à ce que tout soit orthodoxe. Aucune acrobatie : cet accès de purisme m'a même conduite à utiliser des tournures facultatives telles que LET A = 0 où le LET est une pure fioriture. Dans le même ordre d'idées, j'ai tenu à ce que le programme se termine sur l'instruction END (ligne 980) qui, dans le cas présent, est tout à fait superflue.

Par souci de clarté, j'ai également veillé à n'inscrire qu'une seule instruction par ligne lorsque c'était possible, et j'ai séparé les différentes parties du programme en les annon-

çant dans des lignes de remarques (REM) qui sont autant de repères au moment de l'écriture et de la mise au point. Bref, ç'aurait été un véritable devoir de vacances si je ne l'avais fait pour m'amuser.

De tout cela je me suis d'ailleurs bien portée puisque je n'ai pas eu de correction sérieuse à apporter au programme. Je ne parle pas, bien entendu, des broutilles : on est souvent distraite, surtout en voyage, et l'on a tôt fait d'oublier de refermer une parenthèse, mais les petites étourderies de ce genre sont réparées dès qu'un message d'erreur les signale. La conclusion que j'en tire, c'est que l'on a tout à gagner, lorsqu'on écrit un programme, à faire un effort pour qu'il soit clair dès le départ.

C'est l'article d'Alain Perron (pages 32 et 33 du premier numéro de l'Op) qui m'a servi de modèle. Sur TRS 80 Pocket et PC 1211, les règles du Jeu de la Vie sont les mêmes que sur TI 58 et 59. L'univers étudié se compose d'un alignement de dix cases qui peuvent être ou non occupées par un microbe dont la survie dépend du nombre de ses congénères qui l'environnent. Les cases vides, si leur voisinage est favorable, peuvent également se trouver occupées par un nouveau microbe (voir, dans l'encadré, un rappel des règles du Jeu de la Vie).

Règles du Jeu de la Vie, version de Jonathan Millen

Ce jeu, typiquement informatique, consiste à examiner l'évolution d'une population, évolution réglée par quatre lois très simples. La machine se charge des calculs et le joueur, c'est-à-dire la personne qui décide de la première génération (autant dire le dieu) observe comment se comporte sa création.

Dans la version de Jonathan Millen, l'univers est on ne peut plus borné : c'est une rangée de cases alignées (10 cases pour notre programme) qui peuvent être ou non occupées par la vie. L'imagination aidant, on visualisera cette vie sous forme de microbes, et les lois s'énonceront ainsi :

1. Le voisinage d'une case est constitué par les quatre cases qui l'entourent, deux à droite et deux à gauche.
2. Les cases vides dont le voisinage comprend deux ou trois microbes enregistrent une naissance.
3. Les microbes dont le voisinage est composé de quatre cases vides meurent. Il en va de même pour ceux dont le voisinage comprend une ou trois cases occupées.
4. Il ne se produit aucun changement dans les cases qui ne satisfont pas aux conditions des règles 2 et 3.

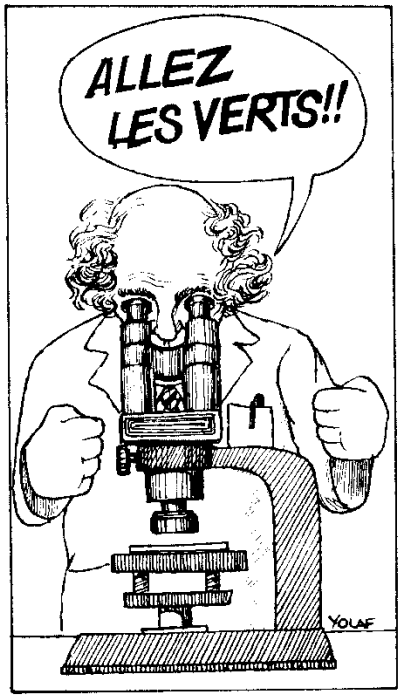
Pour que le jeu ne soit pas faussé par les limites du petit univers étudié, le calcul s'effectue en réalité sur 14 cases, les deux cases de chaque extrémité demeurant par convention toujours vides. Evidemment, on ne visualise que les dix cases du centre puisque ce sont les seules où il peut y avoir quelque chose à observer...

Bien qu'il n'y ait que dix cases " intéressantes ", c'est en réalité 34 registres de mémoires qui sont utilisés pour le calcul des générations et pour leur affichage. Je me suis bien sûr efforcée d'employer en priorité les variables A à Z (autrement dit A (1) à A (26) qui n'empiètent pas sur la zone de mémoire où l'on peut inscrire le programme. Ce sont les mémoires dites " fixes " qui sont toujours à la disposition de l'utilisateur. Les 34 registres employés pour stocker les renseignements utiles à l'état de chaque case se répartissement en trois groupes.

Les 14 premières variables, de A (9) à A (22), autrement dit de I à V, conservent l'état actuel de la population, case par case. Les cases A (9) et A (10), A (21) et A (22), (I et J, U et V), correspondent aux deux extrémités de l'univers : elles sont donc toujours à zéro, c'est-à-dire vides. C'est dans les dix autres variables que l'on inscrit au début de chaque partie un 1 ou un 0 selon que l'on décide que la case correspondante est ou non occupée (lignes 200 à 330 incluses). Comme on le voit sur la liste du programme, un test est effectué à chaque introduction (ligne 230, 300 et 330) de façon à ce qu'il ne soit pas possible d'entrer une valeur différente de zéro ou un.

Un deuxième groupe de dix variables permet de représenter visuellement la génération présente. Il était possible d'en faire l'économie en se contentant d'afficher une suite de zéros et d'uns, mais j'ai pensé qu'on pouvait obtenir facilement une représentation plus agréable à l'œil. C'est la raison pour laquelle, lorsque la première génération est connue, les lignes 410 à 440 rangent dans les variables alphanumériques A$ (23) à A$ (32) un astérisque si la case est occupée et un tiret si elle est vide. C'est ensuite le contenu de ces variables que l'on inscrit à l'affichage ou que l'on imprime sur la CE - 122 grâce à l'instruction PRINT de la ligne 490.

Cette instruction PRINT est décidément bien pratique puisqu'elle permet, sans avoir à le modifier, d'utiliser le programme dans tous les cas, que le TRS 80 Pocket soit ou non connecté à son imprimante. En effet, si l'ordinateur fonctionne de façon autonome, le message demeure à l'affichage jusqu'à ce que la touche ENTER soit enfoncée. On est donc assuré de ne pas le manquer, surtout si l'on a prévu l'émission d'un signal sonore au moyen du BEEP : la machine vous avertit alors qu'elle a quelque chose à vous dire... (1)



Comment votre création se défend-elle ?

(1) Attention toutefois : il ne faut pas trop tarder, car le TRS 80 Pocket comme le PC 1211 s'éteignent d'eux-mêmes après quelques minutes d'inactivité et l'on doit alors tout reprendre depuis le début. L'auto-extinction économise les piles, certes, mais on aimerait parfois que l'ordinateur ne décide pas tout seul du moment où il convient d'interrompre l'exécution d'un programme.

```
10:REM "JEU DES
   MICROBES"
20:REM "AUTEUR
   PAULETTE
   BESNARD"
30:REM "  COPY-
   RIGHT  L^OR-
   DINATEUR DE
   POCHE ET
   L^AUTEUR"
100:REM "INITIA-
   LISATIONS"
110:CLEAR
120:BEEP 1
130:PAUSE "
      BONJOUR..
   ."
200:REM "ENTREE
   DE LA 1ERE
   GENERATION"
210:LET G=1
220:INPUT " PREM
   IERE CASE ?
   ";K
230:IF ((K=0)+(K
   =1))>0)THEN 2
   80
240:BEEP 2
250:PAUSE " ZERO
    OU UN SEULE
   MEMT..."
260:PAUSE " RECO
   MMENCEZ SVP"
270:GOTO 220
280:FOR A=12TO 1
   9
290:INPUT " CASE
    SUIVANTE ?
   ";A(A)
300:IF ((A(A)=0)
   +(A(A)=1))=0
   BEEP 2:PAUSE
   "RECOMMENCEZ
    SVP":GOTO 2
   90
310:NEXT A
320:INPUT " DERN
   IERE CASE ?
   ";T
330:IF ((T=0)+(T
   =1))=0BEEP 2
   :PAUSE " REC
   OMMENCEZ SVP
   ":GOTO 320
400:REM "  AFFI-
   CHAGE DE LA
   GENERATION"
410:FOR A=23TO 3
   2
420:IF (A(A-12))
   LET A$(A)="*
   ":GOTO 440
430:LET A$(A)="-
   "
440:NEXT A
450:BEEP 3
460:IF G<10LET E
   $="   ":GOTO
   490
470:IF G>99LET E
   ="  ":GOTO 49
   0
480:LET E$="   "
490:PRINT E$;G;"
    ";W$;X$;Y$;
   Z$;A$(27);A$
   (28);A$(29);
   A$(30);A$(31
   );A$(32)
500:LET G=G+1
600:REM "LA VIE
   A-T-ELLE
   DISPARU ?"
610:B=K+L+M+N+O+
   P+Q+R+S+T
620:IF B=0THEN 9
   10
700:REM "CALCUL
   DE LA GENE-
   RATION SUI-
   VANTE"
710:FOR A=11TO 2
   0
720:C=A(A-2)+A(A
   -1)+A(A+1)+A
   (A+2)
730:IF ((A(A))*(
   (C=1)+(C=3)+
   (C=0)))LET A
   (A+24)=0:
   GOTO 770
740:IF (A(A))LET
   A(A+24)=1:
   GOTO 770
750:IF ((C=2)+(C
   =3))LET A(A+
   24)=1:GOTO 7
   70
760:LET A(A+24)=
   0
770:NEXT A
800:REM " ACTUA-
    LISATION"
810:FOR A=11TO 2
   0
820:LET A(A)=A(A
   +24)
830:NEXT A
840:GOTO 410
900:REM "LE DE-
   SERT TOTAL"
910:BEEP 4
920:PAUSE " PLUS
    RIEN DE VIV
   ANT !"
930:INPUT " UNE
   AUTRE PARTIE
    ? ";D$
940:IF D$="OUI"
   THEN 210
950:FOR A=1TO 3
960:PAUSE "
    AU REVOIR..
   ."
970:NEXT A
980:END
```

Exemples d'exécution

```
 1. *--***--**
 2. -********--
 3. **-***-**-
 4. -*--*--*-*
 5. --**-**-*-
 6. -*-*-***--
 7. --***-***-
 8. -***---***
 9. **********
10. *-******-*
11. -*--**--*-
12. --**--**--
13. -*--**--*-
14. --**--**--
15. -*--**--*-

 1. *---**-***
 2. --**-*--**
 3. -*-**-**--
 4. --***-*-*-
 5. -***-***--
 6. ***---***-
 7. **********
 8. *-******-*
 9. -*--**--*-
10. --**--**--
11. -*--**--*-
12. --**--**--
13. -*--**--*-
14. --**--**--
```

Liste des variables utilisées	
A (1) ou A	Compteur utilisé pour toutes les boucles : lignes 280-310, 410-440, 710-770, 810-830 et 950-970
A (2) ou B	Nombre de microbes encore en vie : lignes 610 et 620
A (3) ou C	Nombre de microbes vivant dans les quatre cases constituant le voisinage d'une case : ligne 720, 730 et 750
A $ (4) ou D $	Réponse (oui ou non) à l'invitation pour une autre partie : lignes 930 et 940
A $ (5) ou E $	Chaîne de caractères alphanumériques composée d'un, deux ou trois espaces pour la marge gauche de l'affichage : lignes 460 à 490
A (6) ou F	Inutilisée
A (7) ou G	Numéro de la génération affichée : lignes 210, 460 et 470, 490 et 500
A (8) ou H	Inutilisée
A (9) et A (10) ou I et J	Cases de gauche dans la représentation numérique de la génération. Ces deux variables sont toujours à zéro (les cases sont vides) : ligne 720
A (11) ou K	Etat numérique de la première des cases représentées : lignes 220, 230, 420, 610, 720 à 740
A (12) à A (19) ou L à S	Etat numérique des huit cases centrales : lignes 290 et 300, 420, 610, 720 à 740
A (20) ou T	Etat numérique de la dernière des dix cases représentées : lignes 320 et 330, 420, 610, 720 à 740
A (21) et A (22) ou U et V	Etat numérique des deux cases extrêmes à la droite de l'univers. Ces deux variables sont toujours à zéro : lignes 720
A$(23) à A$(32)	Visualisaiton alphanumérique de l'état des dix cases : lignes 420 et 430 à 490.
A (33) à A (44)	Mise en mémoire de la génération à venir : lignes 730 à 760

Inversement, si l'ordinateur est relié à son imprimante, il inscrit son message sur le papier et il poursuit sans attendre. Peu importe alors que l'on ait été là : les écrits restent.

Lorsque l'état des dix cases a été visualisé, le numéro de la génération qui avait été fixé initialement à 1 (ligne 210) est incrémenté de l'unité à la ligne 500 : LET G = G + 1. On vérifie ensuite que la dernière génération compte au moins un microbe (lignes 610 et 620). Si chacune des dix cases est déserte, le jeu s'achève faute de survivant et le programme saute à la ligne 910 pour vous le faire savoir et pour vous proposer une nouvelle partie. Au contraire, s'il reste encore ne serait-ce qu'un microbe, l'exécution se poursuit à la ligne 700 où débute le calcul de la génération suivante.

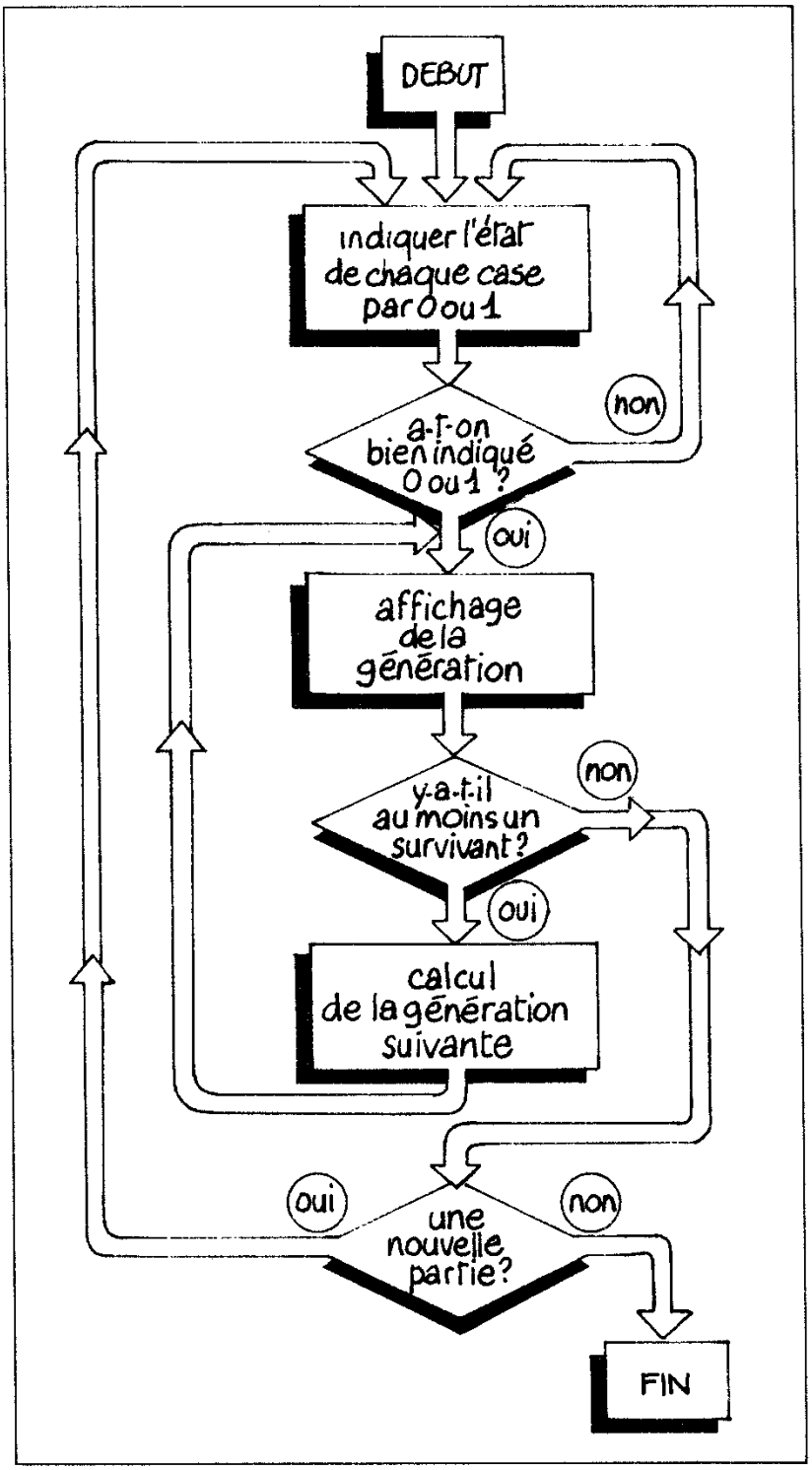
On enregistre alors dans les variables A (33) à A (42) l'état futur des cases en appliquant les règles du Jeu de la Vie.

Le calcul de la génération à venir est effectué aux lignes 710 à 770 et il utilise les fonctions logiques du BASIC de l'ordinateur.

Les lignes 710 à 770 créent une boucle FOR NEXT qui passe en revue, si je peux dire, l'état actuel de chacune des 10 cases.

. La ligne 720 totalise dans la variable C le nombre de microbes présents dans les 4 cases voisines de la case étudiée.

C'est à partir de la ligne 2 (génération actuelle) que l'on calcule la nouvelle génération dont l'image est conservée, au fur et à mesure qu'elle est obtenue, dans la ligne 3. Quand cette phase est terminée, on recopie la ligne 3 dans les dix cases centrales de la ligne 2 : c'est " l'actualisation ". Les cases (9), A(10), A(21) et A(22), autrement dit I, J, U et V, restent toujours à zéro : vides. Reste alors à inscrire dans les variables de la ligne 1 les tirets et les astérisques qui permettront de visualiser la population. ▼



				Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10		
1	Visualisation			W $	X $	Y $	Z $								
				A$ (23)	A$ (24)	A$ (25)	A$ (26)	A$ (27)	A$ (28)	A$ (29)	A$ (30)	A$ (31)	A$ (32)		
				↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		
2	Génération actuelle	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
		A (9)	A (10)	A (11)	A (12)	A (13)	A (14)	A (15)	A (16)	A (17)	A (18)	A (19)	A (20)	A (21)	A (22)
		0	0	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	0	0
3	Génération future			A (33)	A (34)	A (35)	A (36)	A (37)	A (38)	A (39)	A (40)	A (41)	A (42)		

Utilisation du programme

- Rentrer le programme dans la machine (évidemment).
- En mode RUN, faire RUN et presser la touche ENTER.
- A la question " PREMIÈRE CASE ? ", répondre 1 ou 0 (puis presser ENTER) selon que l'on décide que cette case est ou non occupée par un microbe.
- Faire de même jusqu'à la dixième case.
- Si l'imprimante est connectée, la suite des générations s'imprime d'elle-même.
- Dans le cas contraire, on devra appuyer sur ENTER après l'affichage de chaque génération.
- Le programme s'arrête de lui-même si la vie a totalement disparu des dix cases ; il demande alors si l'on désire entamer une autre partie.
- S'il reste ne serait-ce qu'un microbe, il poursuit, et cela peut durer très longtemps. Attention en particulier aux évolutions cycliques.

Questions :

— Y a-t-il des générations telles qu'elles ne puissent pas résulter d'une génération antérieure ?

— Quel est le plus long cycle que l'on peut obtenir ?

— Combien de types d'évolutions cycliques y a-t-il ?

— Avec les règles qui sont appliquées, existerait-il des microbes qui ne meurent jamais ?

P.S. Il ne s'agit pas d'un concours, et je n'ai d'ailleurs pas la réponse aux questions que je vous propose.

. A la ligne 730, si la case est occupée et si son voisinage (variable C) est vide (C = 0) ou comporte 1 ou 3 microbes, elle se vide (LET A (A + 24) = 0), puis on saute à la case suivante : GOTO 770, autrement dit NEXT A.

. Si les conditions de la ligne 730 ne sont pas remplies, on passe à la ligne 740 : la case demeure occupée (LET A(A + 24) = 1), si elle l'était déjà (IF A (A)), et l'on passe à la case suivante.

. Si la case était vide, c'est la ligne 750 qui est exécutée : on fait naître un microbe (LET A (A + 24) = 1) si la voisinage de cette case (C) comporte 2 ou 3 cases occupées, et l'on passe à la case suivante.

. A la ligne 760, enfin, sachant que l'on a épuisé toutes les possibilités, on conclut en décidant que la case demeure vide : LET A (A + 24) = 0 (2).

Et l'on repart pour un tour

Quand l'état des dix cases a été calculé, le programme recopie (lignes 810 à 830) A (33) en A (11), A (34) en A (12), A(35) en A (13), etc. jusqu'à A (42) qui se retrouve en A(20). C'est la phase d'actualisation. On retourne alors en 410 pour distribuer les tirets et les astérisques dans les dix cases à visualiser, et tout est reparti pour un tour...

Afin que l'on puisse toujours savoir où l'on en est, j'ai voulu que le numéro de la génération soit lui aussi affiché, et cela n'a pas été aussi facile à obtenir que je ne le pensais. En effet, de la première à la neuvième génération, ce numéro ne comporte qu'un chiffre ; il en comporte ensuite deux jusqu'à la quatre-vingt-dix-neuvième, puis trois au-delà de la centième. Cela avait pour conséquence de décaler d'une colonne vers la droite l'affichage des populations successives lorsqu'on passait de la 9e à la 10e et de la 99e à la 100e. Sur l'imprimante, l'effet de ce décalage n'était pas très heureux. Je me suis donc arrangée pour que le blanc réservé dans la marge gauche de la bande imprimée (variable alphanumérique E$, E comme ESPace) soit fonction du nombre de chiffres qui composent le numéro de la génération. Lorsque ce numéro ne comporte qu'un chiffre, E$ comprend trois espaces (ligne 460) ; il comprend deux espaces pour les générations de 10 à 99 (ligne 480), et un seul au-delà (ligne 470).

Tout cela — c'est inévitable — occupe de la place dans la mémoire disponible et ralentit légèrement l'exécution du programme, mais il me semble qu'il est bon de soigner autant que possible la présentation. C'est ainsi que, dans tous les messages entre guillemets, j'ai remplacé la lettre O par le chiffre zéro dont le graphisme est nettement meilleur sur l'afficheur et sur l'imprimante.

Comme je l'ai déjà dit, j'ai renuméroté les lignes du programme de 10 en 10. Et pour que la présentation soit encore meilleure, j'ai aussi veillé à ce que chacune de ses parties commence par une ligne de REMarques dont le numéro est un multiple de 100 (100, 200, 400, 600, 700, 800 et 900). Ce sont comme les têtes de chapitre d'un livre. Tout est net. Pour obtenir ce résultat, il m'a fallu, lorsque je devais rajouter une ligne, décaler toute la fin de la liste pour libérer de la place. Cela n'a pas représenté un trop gros travail car le TRS 80 Pocket (même chose bien entendu pour le PC 1211) dispense de recopier la totalité des instructions à déplacer. Cet ordinateur de poche offre en effet une possibilité qu'on trouve sur peu de machines programmables en BASIC : avec lui, il est possible de déplacer une ligne d'instructions en en changeant seulement le numéro. On aurait tort de s'en priver.

Lorsqu'il s'agit de dégager des lignes libres, le plus simple — et de loin — consiste à commencer par la fin. Nous allons voir tout de suite pourquoi. Si le programme que l'on veut modifier compte 10 lignes numérotées de 10 à 100 (10, 20, 30... 90 et 100) et que l'on doit insérer trois lignes après la ligne 80, le programme auquel on aboutira comptera 13 lignes, de 10 à 130. Sa dernière ligne initiale, 100, devra donc s'appeler 130 à l'arrivée.

C'est par elle qu'il convient de commencer : on liste donc la ligne 100 et l'on fait apparaître le curseur de correction (rectangle noir clignotant) sur le numéro de la ligne en utilisant la touche de déplacement du curseur vers la gauche. Il suffit alors d'inscrire le nouveau numéro de ligne (130 dans notre exemple), et de presser sur la touche ENTER. On peut alors supprimer l'ancienne ligne 100, recopiée en 130, en frappant 100 ENTER. Il va de soi qu'au besoin on se ménagera un espace supplémentaire (SHIFT INS) pour inscrire le nouveau numéro de ligne. Dans l'exemple que nous avons pris, la ligne 90 deviendra 120 : on passe donc de deux à trois chiffres.

Voilà, je crois avoir dit l'essentiel. Et je terminerai ce petit article de la même façon que mon programme, en vous disant " au revoir " (ligne 960).

□ Paulette Besnard

(2) Si vous n'êtes pas encore familiarisé avec ces fonctions, mes explications vous sembleront sans doute un peu courtes. Reportez-vous, dans le présent numéro à l'article de la page 51.

Y a-t-il un langage meilleur que les autres ?

Le BASIC est peut-être plus simple, mais il n'est pas certain que cela suffise pour qu'il se taille la part du lion...

■ Je voudrais me faire ici l'avocat des ordinateurs de poche qui n'utilisent pas le langage BASIC, nouveau venu dans le monde des petites machines. A mon avis, le Langage Machine Spécialisée (LMS), qu'il utilise la notation polonaise inverse ou la notation algébrique, a de beaux jours devant lui car il conserve de sérieux atouts. Je ne crois pas que le BASIC constitue pour lui une menace réelle.

Dans le précédent numéro de *l'Op,* André Warusfel a défendu avec conviction les avantages du langage qui — il ne s'en cache pas — a sa préférence. Sa démonstration était intéressante, mais je crois que les arguments qu'il avance méritent d'être revus plus en détail.

Pour commencer, je dirai que les comparaisons portant sur le nombre de lignes n'apportent pas grand-chose : la ligne n'est pas l'unité de compte des ordinateurs. Reprenons l'exemple de la conjecture tchèque (page 20 de *l'Op* n°2). Tel qu'il est écrit, le programme n°1 occupe 19 octets sur HP 67-97 et 23 sur HP 41C. Le programme BASIC n°2 occupe 50 pas (sans compter sa première ligne). Et pour montrer à quel point le nombre de lignes ne reflète pas la taille en mémoire d'un programme, je ferai remarquer qu'on aurait pu écrire le programme n°2 en deux lignes seulement. On supprime la ligne 10 qui est inutile, on conserve la ligne 20 et l'on réécrit les lignes 30 à 70 ainsi :

30 : A = A/2 : A = A+(A−INT A)* (4A+1) : PRINT A : GOTO 30

En ce qui concerne les machines LMS, je propose un autre programme vérifiant la conjecture tchèque et conçu pour les HP 41 et HP 67-97.

HP 41	HP 67-97
	LBL A
2	
ENTER ↑	
ENTER ↑	
RND	
x = y ?	
CHS	
+	
+	
END	RTN

Comme vous le voyez, il est spécialement court et ne demande qu'un peu plus de matière grise. Il s'utilise en FIX 0 ou DSP 0. Nous reviendrons sur ce programme au moment de tirer les conclusions.

Y a-t-il un langage meilleur que les autres ?

Concernant l'élévation de matrice à la puissance 5, même remarque sur l'occupation de la mémoire : le programme BASIC n°4 (15 lignes) occupe en réalité 112 pas sur PC 1211, sans compter les 30 pas de la ligne 10 dont on peut effectivement se passer. Signalons aussi que " l'affichage final ponctué de RUN/STOP en coups de cymbales " sur les HP présente un avantage certain : il donne au moins les résultats en clair, ce qui n'est pas toujours le cas du programme BASIC proposé. Comment interpréter en effet :
1702.68752025.28125675.0
qui représente le résultat obtenu à l'affichage de la PC 1211 avec la matrice :

$$M = \begin{bmatrix} 3 & 4,5 \\ 1,5 & 1 \end{bmatrix}$$

On est donc également conduit à lire les résultats " avec coups de cymbales " puisqu'on doit demander (ou programmer) :

E	ENTER	affichage	1702.6875
F	ENTER	affichage	2025.28125
G	ENTER	affichage	675.09375
H	ENTER	affichage	802.5625

Ecrire en BASIC, dit-on, c'est plus simple. Et il est sans doute vrai que c'est un peu plus " confortable ", mais les programmes n'y gagnent rien en rapidité comme le montrent les résultats du petit test que j'ai effectué.

Pour poursuivre la comparaison, j'ai pris un troisième exemple où les calculatrices du type TI ou HP sont moins à l'aise que les ordinateurs : le classement des données (Programme H 16, page 266 du manuel d'applications de la PC 1211).

classement de données	encombrement prgm oct.	reg.	total reg.	temps de classement de 35 données	maximum de données admissibles
HP 41 prgm n°1	44	0	7	1 m 30 s	56 à 312 suivant
prgm n°2	95	1	15	1 m 05 s	48 à 304 le nombre de modules RAM
PC 1211	82	4	15	6 m 54 s	189

Sur PC 1211, le programme testé est une version adaptée de celui qui est proposé dans le manuel :

```
10 : " C " AREAD D : FOR C = 5
     TO D + 3
20 : FOR A = C + 1 TO D + 4
30 : IF A (C) > = A (A) THEN 50
40 : B = A (C) : A (C) = A (A) : A
     (A) = B
50 : NEXT A : NEXT C : END
```

Tous ces exemples montrent que la rapidité du BASIC Sharp n'est pas d'une évidence aveuglante.

Cela étant dit, je tiens à préciser que je ne suis pas un adversaire du BASIC. J'ai tenu seulement à présenter la comparaison sous un autre angle. Il ne suffit pas qu'un langage soit dit évolué pour qu'il soit supérieur à tout point de vue. Cela est si vrai que les programmes performants qui " tournent " sur les (gros) ordinateurs sont souvent écrits en langage machine. A cet égard, le LMS constitue donc un compromis acceptable, et d'ailleurs accepté, je pense l'avoir assez bien montré.

Il oblige à une meilleure analyse du problème *et* à une meilleure connaissance des ressources de la calculatrice pour adapter la formulation mathématique aux moyens disponibles. Le goût du travail " léché " est *l'une* des satisfactions que s'offre le programmeur en LMS : c'est une tournure d'esprit.

On peut évidemment faire valoir qu'en s'initiant d'entrée de jeu au BASIC sur un ordinateur de poche, on sera mûr pour passer au stade supérieur : l'ordinateur de table ou de bureau.

Mais cela mérite qu'on y réfléchisse car il n'est pas certain que tout le monde saute le pas :

. les xxxpoches sont portatifs, et c'est une de leurs principales qualités

. les machines plus encombrantes sont plus chères aussi, et cela peut être un obstacle

. enfin et surtout, les ordinateurs de bureau ne rendent pas les mêmes services. Pour la plupart, ils sont assez mal équipés pour le calcul scientifique : il leur manque de nombreuses fonctions préprogrammées d'usage courant qui figurent sur le clavier des xxxpoches :

. P→R, R→P, D→R, R→D pour les mathématiciens, physiciens, électroniciens, géomètres, astronomes et navigateurs ;

. fonctions statistiques

. drapeaux, etc.

Voilà ce que l'avocat de la défense avait à faire entrer dans la Mémoire Vive des lecteurs afin que chacun puisse choisir son matériel en fonction des problèmes qu'il a à résoudre. Et, paraphrasant les termes mêmes d'André Warusfel, parlant des machines LMS, je conclurai en disant : " Fréquentons-les comme de vieux amis, et pensons aux longues heures de loisir *intelligent* qu'elles nous procurent ".

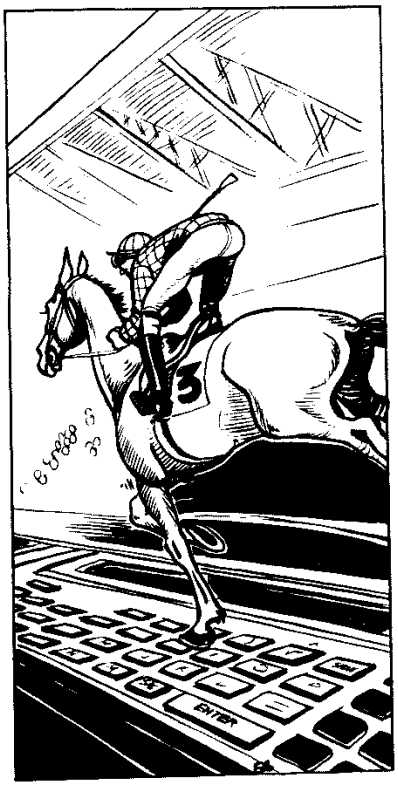
☐ Jean Thiberge

conjoncture tchèque			
	encombrement	temps	
HP 41	12 octets	27 s	Le temps a été mesuré en boucle depuis N = 41 jusqu'à obtention de N = 1
PC 1211	50 pas	52 s	
(prgm A)	38 pas	44 s	A : avec ligne 30 proposée ci-avant
matrice			
HP 41	61 octets	≃ 4 s	Proramme n°3 sans autre modification que la rectification de l'entrée des données.
PC 1211	112 pas	≃ 6 s	

vie pratique

Ne perdez plus votre temps pour jouer au tiercé ou au loto

Jouer au loto ou jouer au tiercé est un sport national. Jouons donc mais pourquoi ne pas être aidé par une TI-59 ? Elle calculera à votre place diverses probabilités et vous donnera — peut-être ! — quelques gagnants. Prévenez-nous si vous gagnez !



■ Sur un xxxpoche, le loto n'est certainement pas plus facile. Il est moins cher, mais il ne rapporte rien. Le seul avantage du programme proposé ici, c'est qu'il n'attend pas le mercredi soir pour vous donner les numéros gagnants. Inutile d'attendre une semaine pour apprendre qu'on a encore perdu : le tirage ne demande qu'une minute. Quant au tiercé, il peut maintenant (sur le même xxxpoche et grâce au même programme) se courir à domicile avec les favoris et les tocards.

Les courses de chevaux : pour les paris simples, couplés, tiercés, etc. on associe à chaque cheval son coefficient de chance. L'ordre d'arrivée n'est pas une pure loterie : plus les chances d'un cheval sont grandes, plus la probabilité de le voir arriver dans les premiers augmente. Bref, la course se déroule exactement comme elle devrait se dérouler dans la réalité... si les pronostiqueurs étaient infaillibles. Pour une course à chances égales, il suffit d'associer à chaque cheval des probabilités égales (1 par exemple).

Le loto : pour simuler le loto, il faut bien entendu que chaque boule ait autant de chances de sortir que les autres. Si, par erreur ou par malice, vous avez associé au numéro 48 une probabilité de 2 en conservant 1 pour toutes les autres boules, vous avez pipé le jeu : il y a dans le sac deux boules n° 48. Vous allez être soupçonné.

Le programme suivant exécute une simulation du tirage du loto et des courses de chevaux. Les chances de chaque cheval sont prises en compte.

Après avoir entré le programme — groupe 1 et 2 — en position normale 479.59 il vous faut appliquer le mode d'emploi suivant :

1 - introduire un numéro (cheval ou boule de loto) au clavier (appuyer sur *A*) : l'ancienne probabilité associée à ce numéro est affichée ;

Exemple d'exécution

```
  1.        96.
160.        13.
            10.
  2.
 50.
                14.
  3.        10.
   2.        6.

  4.         5.
150.        10.
             6.
  5.
 55.         7.
             4.
  6.
 20.

  7.         7.
 35.        15.

  8.        14.
 40.        13.
             9.

  9.         5.
 10.        12.
             7.

 10.
110.
                6.
 11.            6.
  6.            10.

 12.            15.
 55.            14.

 13.            12.
 40.            17.
                15.

 14.            14.
 96.            20.

 15.            16.
 63.
```

2 - introduire la nouvelle probabilité (1 à 9999) et presser sur *R/S*, répéter l'opération 1 et 2 **au moins** autant de fois qu'il y a de numéros sur une grille ou de chevaux pariés ;

3 - mettre en forme l'ensemble des probabilités rentrées en 1 et 2 en appuyant sur *B* (**obligatoire**, avant l'opération suivante) ;

4 - initialisation des tirages : appuyer sur *E* ;

5 - introduire un nombre (de 1 à 301299) pour amorcer la génération des nombres aléatoires puis appuyer sur *R/S* ;

6 - introduire le nombre de numéros tirés sur une grille ou le nombre de

Liste du programme sur TI-59

000	10	E'	086	52	EE	172	65	65	236	99	PRT	300	42	STO	
001	76	LBL	087	42	STO	173	73	RC*	237	55	÷	301	62	62	
002	42	RCL	088	65	65	174	00	00	238	04	4	302	42	STO	
003	43	RCL	089	06	6	175	65	×	239	32	INV	303	63	63	
004	60	60	090	06	6	176	43	RCL	240	28	LOG	304	32	X:T	
005	42	STO	091	44	SUM	177	04	04	241	95	=	305	43	RCL	
006	11	11	092	04	04	178	95	=	242	73	RC*	306	08	08	
007	36	PGM	093	73	RC*	179	44	SUM	243	00	00	307	22	INV	
008	15	15	094	04	04	180	65	65	244	46	ADV	308	77	GE	
009	13	C	095	55	÷	181	43	RCL	245	41	R/S	309	10	E'	
010	59	INT	096	43	RCL	182	65	65	246	10	E'	310	41	R/S	
011	32	X:T	097	65	65	183	59	INT	247	76	LBL	311	99	PRT	
012	43	RCL	098	95	=	184	65	×	248	15	E	312	42	STO	
013	61	61	099	22	INV	185	73	RC*	249	46	ADV	313	64	64	
014	42	STO	100	59	INT	186	00	00	250	46	ADV	314	46	ADV	
015	11	11	101	55	÷	187	95	=	251	25	CLR	315	46	ADV	
016	03	3	102	93	.	188	72	ST*	252	07	7	316	76	LBL	
017	04	4	103	01	1	189	00	00	253	69	OP	317	33	X²	
018	42	STO	104	49	PRD	190	69	OP	254	17	17	318	71	SBR	
019	00	00	105	65	65	191	20	20	255	43	RCL	319	43	RCL	
020	73	RC*	106	95	=	192	97	DSZ	256	08	08	320	22	INV	
021	00	00	107	59	INT	193	07	07	257	29	CP	321	67	EQ	
022	22	INV	108	32	X:T	194	01	01	258	67	EQ	322	00	0	
023	77	GE	109	00	0	195	73	73	259	10	E'	323	43	RCL	
024	00	00	110	92	RTN	196	43	RCL	260	09	9	324	65	65	
025	30	30	111	76	LBL	197	11	11	261	09	9	325	74	SM*	
026	01	1	112	12	B	198	42	STO	262	09	9	326	04	04	
027	01	1	113	25	CLR	199	61	61	263	09	9	327	43	RCL	
028	42	STO	114	07	7	200	06	6	264	42	STO	328	07	07	
029	03	3	115	69	OP	201	69	OP	265	65	65	329	37	IFF	
030	05	5	116	17	17	202	17	17	266	69	OP	330	07	07	
031	44	SUM	117	25	CLR	203	91	R/S	267	07	07	331	01	01	
032	00	00	118	42	STO	204	10	E'	268	69	OP	332	34	√x	
033	73	RC*	119	04	04	205	76	LBL	269	16	16	333	91	R/S	
034	00	00	120	42	STO	206	11	A	270	33	X²	334	99	PRT	
035	22	INV	121	08	08	207	99	PRT	271	43	RCL	335	97	DSZ	
036	77	GE	122	04	4	208	32	X:T	272	11	11	336	00	00	
037	00	00	123	09	9	209	00	0	273	67	EQ	337	03	03	
038	30	30	124	42	STO	210	42	STO	274	00	00	338	25	CLR	
039	05	5	125	07	07	211	08	08	275	76	76	339	42	STO	
040	22	INV	126	01	1	212	77	GE	276	42	STO	340	66	66	
041	44	SUM	127	01	1	213	10	E'	277	61	61	341	42	STO	
042	00	00	128	42	STO	214	04	4	278	25	CLR	342	67	67	
043	73	RC*	129	00	00	215	09	9	279	42	STO	343	42	STO	
044	00	00	130	29	CP	216	22	INV	280	10	10	344	68	68	
045	69	OP	131	73	RC*	217	77	GE	281	42	STO	345	42	STO	
046	20	20	132	00	00	218	10	E'	282	66	66	346	69	69	
047	22	INV	133	22	INV	219	32	X:T	283	42	STO	347	98	ADV	
048	77	GE	134	59	INT	220	85	+	284	67	67	348	43	RCL	
049	00	00	135	73	ST*	221	01	1	285	42	STO	349	67	67	
050	43	43	136	00	00	222	00	0	286	66	66	350	43	RCL	
051	69	OP	137	67	EQ	223	95	=	287	42	STO	351	62	62	
052	30	30	138	01	01	224	42	STO	288	69	69	352	97	DSZ	
053	43	RCL	139	43	43	225	00	00	289	69	OP	353	04	4*	
054	00	00	140	69	OP	226	73	RC*	290	07	07	354	33	X²	
055	75	-	141	38	38	227	00	00	291	91	R/S	355	06	6	
056	01	1	142	66	NOP	228	22	INV	292	99	PRT	356	69	OP	
057	00	0	143	44	SUM	229	59	INT	293	55	÷	357	17	17	
058	95	=	144	04	04	230	65	×	294	02	2	358	98	ADV	
059	42	STO	145	69	OP	231	04	4	295	95	=	359	48	EXC	
060	04	04	146	20	20	232	22	INV	296	42	STO	360	91	R/S	
061	42	STO	147	97	DSZ	233	28	LOG	297	09	09	361	19	E'	
062	07	07	148	07	07	234	95	=	298	91	R/S	362	00	0	
063	55	÷	149	01	01	235	91	R/S	299	99	PRT	363	00	0	
064	01	1	150	31	31										
065	03	3	151	04	4										
066	95	=	152	09	9										
067	59	INT	153	42	STO										
068	48	EXC	154	07	07										
069	04	04	155	01	1										
070	75	-	156	01	1										
071	01	1	157	42	STO										
072	03	3	158	00	00										
073	65	×	159	01	1										
074	43	RCL	160	00	0										
075	04	04	161	00	0										
076	95	=	162	00	0										
077	94	+/-	163	02	2										
078	85	+	164	55	÷										
079	01	1	165	43	RCL										
080	04	4	166	04	04										
081	95	=	167	95	=										
082	22	INV	168	42	STO										
083	28	LOG	169	04	04										
084	52	EE	170	00	0										
085	22	INV	171	42	STO										

Etiquette A (pas 205) : chargement des probabilités ; l'ancienne probabilité est rappelée à l'affichage aux pas 234 et 235.
Etiquette B (pas 111) : mise en forme des probabilités.
Etiquette RCL (pas 002) : tirage des résultats ; le sous-programme de génération des nombres aléatoires (contenu dans le module de base) est appelé aux pas 007 à 009.
Etiquette E (pas 247) : programme principal.

chevaux à l'arrivée, et de nouveau *R/S* (après exécution, rappel du nombre de probabilités rentrées) ;
7 - introduire le nombre de tirages ou de courses souhaités, enfin touche *R/S*.

Résultats : la connexion de l'imprimante est détectée automatiquement. Si elle est connectée, les nombres tirés s'inscrivent successivement et chaque groupe de résultats est séparé du suivant par un saut de ligne. Dans le cas contraire, après l'affichage d'un nombre, on presse *R/S* pour obtenir le suivant. A la fin des résultats, on retrouve la partition initiale 479.59.

Rejouez vos favoris immédiatement

En cas de fausse manœuvre, l'affichage clignote : on a sans doute oublié de mettre en forme les probabilités (touche *B*). Si ce n'est pas le cas, le nombre des numéros à tirer dépasse certainement celui des probabilités rentrées : il ne peut pas y avoir plus de chevaux à l'arrivée qu'au départ !

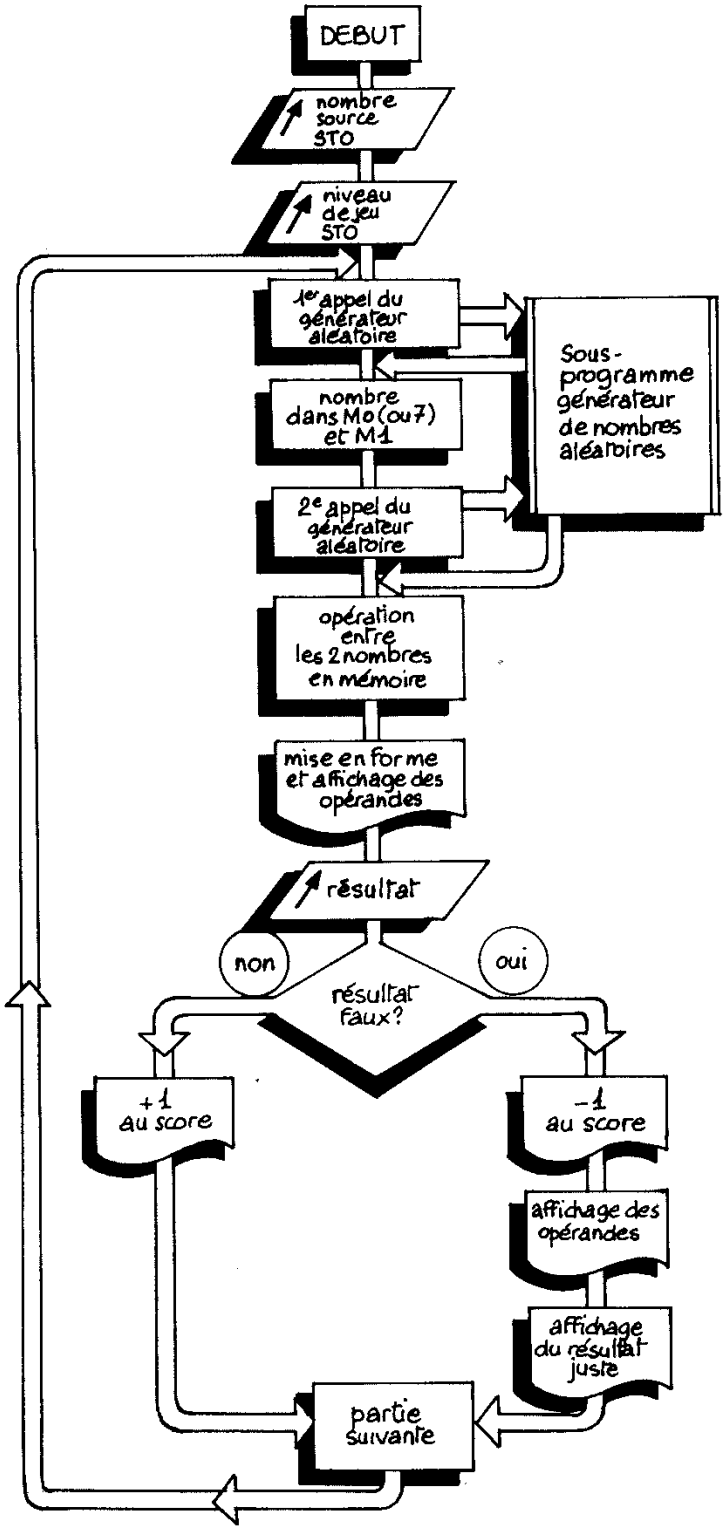
Il est possible d'enregistrer sur fiche magnétique non seulement le programme, mais les probabilités (groupes 3 et 4). Les probabilités des numéros 1 à 49 sont respectivement conservés dans les mémoires 11 à 59. Pour le listage des mémoires sur le PC 100, on utilisera la fonction O INV 2nd List. Si l'on décide de changer la probabilité de tel ou tel numéro, il faudra exécuter *B* avant de relancer le programme proprement dit avec la touche *E*.

Un dernier mot : les amateurs de martingales préparent souvent leur grille de loto en fonction des numéros sortis lors des tirages précédents. Avec ce programme, ils pourront choisir leur grille en favorisant les numéros dont ils estiment qu'ils ont plus de chances de sortir que les autres. Bonne chance...

☐ Dominique Coupleux

récréation arithmétique

Quand votre micropoche vous pose des colles



Ordinairement, la calculatrice exécute les opérations que vous lui demandez. Elle peut également vous demander d'en faire...

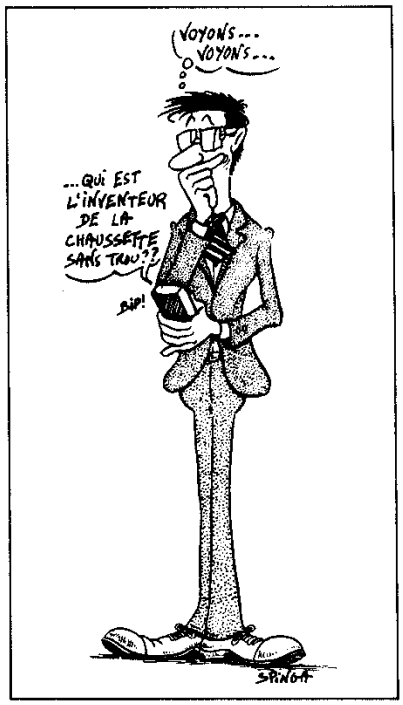
■ L'utilisation d'une calculatrice programmable ne dispense pas de la pratique du calcul mental. Mais peut-être vous sentez-vous engourdi après toutes ces heures passées sur votre clavier. Donnez donc aujourd'hui à votre ordinateur de poche l'occasion de se faire pardonner en essayant ce petit programme.

Un algorithme simple

Pas de complication du côté de l'algorithme. Il suffit de faire produire puis afficher par la calculatrice deux nombres tirés au hasard dont il faudra calculer la somme, la différence, le produit ou le quotient à votre choix. La machine fera elle aussi l'opération et elle comparera les deux résultats. Comme elle est vraiment à son aise dans cette matière, elle considèrera toujours qu'elle a raison et elle vous attribuera un mauvais point si votre réponse est différente de la sienne, mais elle vous fournira alors le résultat juste.

L'organigramme détaille la suite des opérations : cela commence par un préliminaire pendant lequel on introduit un nombre source pour le générateur de nombres aléatoires, puis le niveau de jeu, c'est-à-dire le nombre maximum de chiffres des opérandes. Ensuite le programme principal appelle le générateur de nombres aléatoires qui fournit le premier opérande. Ce nombre est stocké successivement dans deux mémoires. La première servira à la calculatrice pour effectuer l'opération, la seconde sera utilisée pour le façonnage et la présentation simultanée des deux opérandes à l'affichage sous la forme O_1, O_2, ce qui facilitera la lecture. Mais auparavant

Quand votre micropoche vous pose des colles



le générateur aléatoire aura fourni le deuxième nombre.

Une fois l'opération effectuée, ce deuxième nombre est divisé par 10^n (n étant le niveau de jeu) et il est ajouté au premier opérande. Cette représentation conjointe des deux opérandes est affichée le temps d'une pause. Un zéro apparaît alors et l'exécution s'arrête. Le joueur introduit le résultat qu'il a calculé et la machine le compare avec celui qu'elle a obtenu.

S'il y a désaccord entre les deux, le score du joueur diminue d'un point et les deux opérandes réapparaissent, suivis par le résultat juste. Si les deux nombres sont identiques, la calculatrice ajoute un point et la partie suivante recommence aussitôt.

Le sous-programme générateur de nombres aléatoires n'a pas été détaillé dans l'organigramme. Il est très classique : le nombre source est multiplié par 67, puis la partie fractionnaire est isolée et mise en mémoire pour servir de nombre source dans le prochain tirage. On multiplie ensuite par 10^n (n, g 10^x, x sur HP 33 ou X, n, INV 2nd Log = sur TI 57), n étant le niveau de jeu, c'est-à-dire le nombre maximum de chiffres que comporte le nombre produit. Il ne reste plus qu'à couper la partie fractionnaire (g INT ou 2nd INT) pour avoir l'un des deux opérandes utiles au programme principal.

Entrez le programme correspondant à votre calculatrice puis faites g RTN sur HP 33 ou RST sur TI 57. Introduisez un nombre source compris entre 0 et 1 (exemple : 0,3852176) comprenant le plus de chiffres possible pour que le générateur aléatoire ne déraille pas. Faites R/S et introduisez le niveau du jeu (de 1 à 4, ou même 5 sur HP 33, mais il ne faut pas être présomptueux...). Un nouveau R/S démarre la première partie et les deux opérandes sont affichés le temps d'une pause de part et d'autre du point décimal.

Si le niveau est 1 vous obtiendrez par exemple " 8.7 ". Additionnez mentalement et entrez le résultat : 15, puis faites R/S. Vous avez gagné puisque la calculatrice vous

Programme de calcul mental, version HP 33

01	23 3	STO 3	— mise en mémoire du nombre source
02	74	R/S	— arrêt ; entrée du niveau de jeu
03	23 2	STO 2	— mise en mémoire
04	12 33	GSB 33	— appel du générateur aléatoire
05	23 0	STO 0	— mise en mémoire
06	23 1	STO 1	
07	12 33	GSB 33	— 2e appel du générateur aléatoire
08	23 51 0	STO + 0	— addition des deux nombres
09	24 2	RCL 2	mise en forme pour affichage des opérandes
10	15 2	g 10^x	
11	71	÷	
12	23 51 1	STO + 1	
13	24 1	RCL 1	affichage
14	14 74	f PAUSE	
15	34	CLx	— remise à zéro de l'affichage
16	74	R/S	— entrée du résultat
17	24 0	RCL 0	le résultat est-il faux ? si oui aller à 24, sinon pas suivant :
18	14 61	f x ≠ y	
19	13 24	GTO 24	
20	1	1	1 additionné au score affichage du score partie suivante
21	23 51 4	STO + 4	
22	14 74	f PAUSE	
23	13 04	GTO 04	
24	1	1	— 1 additionné au score affichage rappel des opérandes affichage rappel du résultat affichage partie suivante
25	32	CHS	
26	23 51 4	STO + 4	
27	14 74	f PAUSE	
28	24 1	RCL 1	
29	14 74	f PAUSE	
30	24 0	RCL 0	
31	14 74	f PAUSE	
32	13 04	GTO 04	
33	24 3	RCL 3	sous-programme générateur de nombres aléatoires
34	6	6	
35	7	7	
36	61	x	
37	15 33	g FRAC	
38	23 3	STO 3	
39	24 2	RCL 2	
40	15 2	g 10^x	
41	61	x	
42	15 32	g INT	
43	15 12	g RTN	

donne 1 point avant de repartir pour l'opération suivante. Si vous aviez introduit par exemple 16 (tout le monde peut se tromper), vous auriez obtenu un mauvais point : −1 et vous auriez pu relire les deux opérandes, 8.7, suivis du résultat juste (15) avant que la partie ne continue. Pour consulter votre score, vous pouvez rappeler la mémoire n°4 pendant toute interruption du programme.

Il faut bien se rappeler que la virgule flottante n'existe pas sur la HP 33. On doit donc adapter le format d'affichage au niveau de jeu avec f Fix n (n = niveau) et penser à le modifier à chaque changement de niveau.

00	32 3	STO 3	— mise en mémoire du nombre source
01	81	R/S	— arrêt, entrée du niveau de jeu
02	32 2	STO 2	— mise en mémoire
03	86 0	2nd Lbl 0	— étiquette du programme principal
04	61 3	SBR 3	— appel du générateur aléatoire : 1er nombre
05	32 7	STO 7	— mémoire pour le calcul
06	32 1	STO 1	— mémoire pour l'affichage
07	61 3	SBR 3	— appel générateur : 2e nombre
08	34 7	SUM 7	— (ou INV SUM 7 ou Prd 7) - opération en mémoire
09	45	÷	
10	33 2	RCL 2	mise en forme pour affichage des opérandes
11	−18	INV 2nd Log	
12	85	=	
13	34 1	SUM 1	
14	33 1	RCL 1	affichage
15	36	2nd Pause	
16	15	CLR	
17	81	R/S	— entrée du résultat
18	−66	INV 2nd x=t	— est-il faux ?
19	51 1	GTO 1	— si oui aller à l'étiquette 1
20	01	1	— si non, ajouter 1 au score
21	34 4	SUM 4	
22	36	2nd Pause	— afficher le point
23	51 0	GTO 0	— partie suivante
24	86 1	2nd Lbl 1	quand le résultat est faux
25	01	1	−1
26	84	+/−	
27	34 4	SUM 4	— additionné au score
28	36	2nd Pause	— et affiché
29	33 1	RCL 1	— rappel des opérandes
30	36	2nd Pause	— et affichage
31	33 7	RCL 7	— rappel du résultat
32	36	2nd Pause	— et affichage
33	51 0	GTO 0	— partie suivante
34	86 3	2nd Lbl 3	
35	33 3	RCL 3	
36	55	x	
37	06	6	
38	07	7	
39	85	=	
40	−49	INV 2nd Int	sous-programme générateur de nombres aléatoires
41	32 3	STO 3	
42	55	x	
43	33 2	RCL 2	
44	−18	INV 2nd Log	
45	85	=	
46	49	2nd Int	
47	−61	INV SBR	

Choisissez votre opération

Tel qu'il est présenté, le programme additionne les deux nombres, mais vous pouvez changer les opérations en modifiant le pas 08 par STO−0, STO×0 sur HP 33, INV SUM 7 ou 2nd Prd sur TI 57. La division est également réalisable à condition de rajouter un test qui élimine les zéros que le générateur de nombres aléatoires peut produire de temps à autre. Cela dit, les décimales ne facilitent pas le jeu et peuvent même le fausser à cause des chiffres de garde qui n'apparaissent pas à l'affichage.

Quand les opérandes comportent plus de trois chiffres, il devient indispensable, pour avoir le temps de les lire, de remplacer la pause d'affichage par un R/S.

Enfin, pour adapter le programme à d'autres types d'ordinateurs de poche, il n'y aura pas de grosses difficultés. Sur TI 58-59, on remplacera les étiquettes numériques de la TI 57 par des labels utilisateurs, et on modifiera l'accès au registre t qui n'est pas direct.

Sur HP 34 ou 67, il faut en partant de la liste pour HP 33 placer des labels à l'arrivée des branchements et les adresser convenablement.

Avec les HP 41 et PC 1211, il est possible d'égayer le jeu grâce à des messages alphanumériques. Avec les drapeaux de la HP 41, j'ai même essayé une version qui choisit au hasard l'opération à réaliser et qui l'indique au joueur au moyen des petits chiffres qui indiquent l'état des quatre premiers drapeaux. On peut même imaginer d'autres améliorations comme celle qui consiste à faire afficher en clair le signe de l'opération entre les deux nombres proposés en utilisant la fonction *Append*.

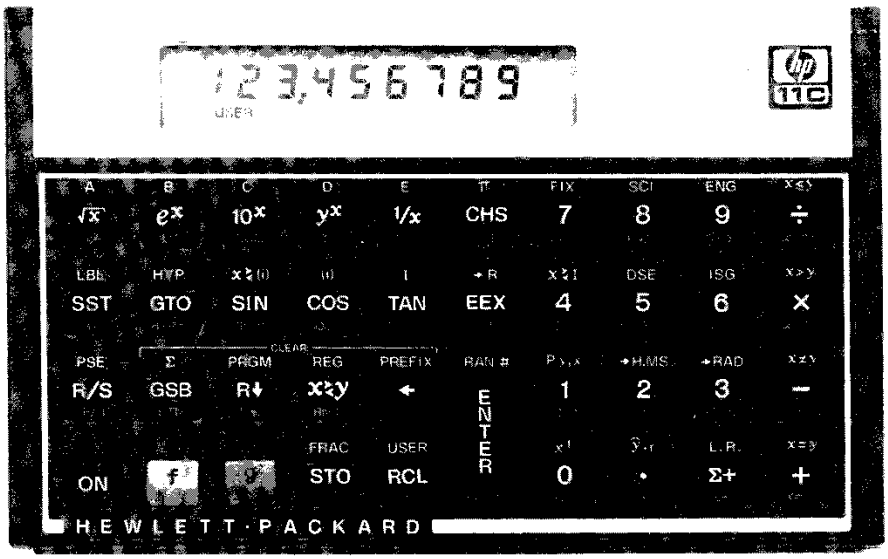
Quoi qu'il en soit, ce programme est, je l'espère, de nature à redonner du goût pour le calcul mental. On entend dire parfois qu'avec les calculatrices les gens ne sauront plus compter. Je pense au contraire que l'emploi de ces machines contribuera à nous y exercer.

☐ Xavier de La Tullaye

Coup d'œil sur...

Le nouveau modèle de HP

L'un des deux derniers-nés de Hewlett-Packard est un scientifique. Nous découvrons le HP 11C



■ Dans son étui souple (simili-cuir noir), on pourrait le prendre, si l'on est distrait, pour un petit portefeuille qui ne serait pas trop épais : le nouveau micropoche de Hewlett-Packard (110 grammes si j'en crois mon pèse-lettre) est vraiment fait pour la poche. Son épaisseur est de seize millimètres...

Quand on l'extrait pour la première fois de son étui, on a une surprise : HP nous avait habitués à des machines plus hautes que larges, mais celle-ci a été conçue pour une utilisation horizontale. Il faudra voir à l'usage si cette nouvelle disposition constitue ou non un avantage.

En tout cas, le premier coup d'œil me donne l'impression d'une machine assez futuriste (je ne parle que du design) : métal brossé pour le bandeau supérieur qui comporte l'afficheur à cristaux liquides et fond noir pour le clavier où les touches, noires aussi, portent des inscriptions en blanc, bleu et jaune. Les touches ont toutes leur face avant en pan coupé comme sur la 41 CV et sont agréables au doigt. A l'enfoncement, le déclic est doux, mais les documents qui accompagnent la machine nous affirment que chacune des touches a été testée pour survivre à 150 000 pressions, et cela me paraît être plus que suffisant.

L'afficheur à cristaux liquides fait plaisir à voir avec ses grands chiffres noirs : de quoi faire rougir de honte les diodes des grands-mères HP. Cela étant dit, il n'occupe que 6 centimètres de large et il n'est pas très bavard. Mis à part les messages du type " Error " ou " Running ", il est exclusivement numérique, et cela fait un peu chiche quand on s'est habitué à celui des 41C. Au bas de l'affichage, des indicateurs signalent l'état des piles, l'appel des touches f et g, le mode radian ou grade, user ou programme, mais je regrette que l'affichage ne dise pas si le programme attend l'entrée d'une donnée et n'indique pas l'état des drapeaux.

De plus en plus souvent, les " cristaux liquides " vont de pair avec la mémoire permanente, et on le vérifie ici encore une fois : la mémoire conserve bien les données et les programmes, les modes angulaires, etc. C'est devenu indispensable sur une calculatrice récente.

A l'allumage de la machine, vous avez le choix entre la notation anglo-saxonne, c'est-à-dire le point décimal et la virgule pour séparer les chiffres par groupe de trois (19, 637, 421.95 par exemple), et la notation européenne qui consiste à faire le contraire (19.637.421,95). Tous les calculs s'effectuent, comme on pouvait s'y attendre, en notation polonaise inverse. Chacun a ses raisons pour préférer la notation à laquelle il s'est accoutumé, et pour ne fâcher personne, je dirai que la notation polonaise inverse est très commode : elle permet en particulier des tas d'astuces de programmation, mais, tout bien pesé, les partisans de la notation algébrique ont également leurs arguments et, comme toujours, chacun verra midi à sa porte.

Dans les bons points à ne pas oublier, il faut mentionner la correction pas à pas (flèche arrière, comme sur la 41 C) du nombre en cours d'introduction. C'est tellement plus pratique que le seul CLx qui oblige à tout retaper (1).

Le passage du mode calcul-exécution au mode programme est confié à une touche standard du clavier (comme l'allumage d'ailleurs), ce qui est nouveau chez ce constructeur.

Au chapitre des fonctions mathématiques et scientifiques, il ne faut vraiment pas se plaindre : on est comblé. Je ne les citerai pas toutes, vous pouvez les lire sur la photo (grandeur nature). Je me contenterai de mentionner les plus remarquables : fonctions hyperboliques, factorielles et fonction gamma (factorielles généralisées), permutations et combinaisons, pourcentages et écarts de pourcentages. Plus original : un générateur de nombres aléatoires avec conservation par la mémoire permanente d'un nombre-

(1) Très bizarrement, cette touche de correction n'existe pas sur la calculatrice programmable financière HP 12C introduite sur le marché en même temps que la 11C : les financiers et les comptables taperaient-ils moins de chiffres que les scientifiques ?

source. Une petite déception toutefois : j'ai cherché en vain la fonction SOLVE qui fait le bonheur des matheux lorsqu'ils utilisent une HP 34.

A noter encore, pour l'utilisation des fonctions statistiques et autres, un aide-mémoire gravé au dos de la machine. Il est bien fait, assez complet, et donc très utile.

En résumé, le lot des fonctions est l'un des plus étendus qu'on puisse trouver sur une machine de ce type, le tout étant organisé dans une syntaxe qui est devenue familière à tout possesseur d'une précédente HP.

En ce qui concerne la programmation, on ne sera pas non plus désemparé. On passe des HP 34 ou 67 à la 11C les yeux presque fermés. Je dis " presque " car, si les principes sont identiques, la disposition différente des touches et le changement des numéros de code bousculent les habitudes acquises. Cela dit, on s'y fera au prix d'un petit effort d'adaptation.

Les codes des touches sont donc nouveaux, mais leur expression dans la liste des programmes reste malheureusement numérique. C'est dommage de ne pas avoir profité des possibilités offertes par les cristaux liquides pour écrire les codes sous forme de lettres comme sur la 41C. On aurait aimé avoir au moins une indication du préfixe en clair, même un afficheur exclusivement numérique peut la fournir.

Voyons maintenant ce qu'il en est de la mémoire réservée au programme. Je vous préviens, si vous êtes très sensible à cet aspect des choses, vous pouvez sortir vos mouchoirs, car je vous annonce 21 registres de données et 63 lignes de programme dont chacune contient une instruction, que cette instruction soit obtenue par la pression d'une, deux ou trois touches. C'est ainsi que l'étiquette E par exemple (touches f LBL f E, que l'on obtient également en abrégé grâce à f LBL E) n'occupe qu'une seule ligne de programme.

Or 63 lignes, c'est beaucoup mieux que 40, mais pour une machine qui a de telles possibilités dans les fonctions mathématiques, c'est un peu maigre. On le regrettera d'autant plus que la mémoire est permanente et qu'on dispose de touches utilisateur (on en reparlera d'ailleurs, de ces fameuses touches...).

Pour être tout à fait complet, il faut ajouter que la mémoire globale peut se répartir entre programme et données. Il est donc possible de renoncer à des registres de stockage pour les convertir en lignes de programme supplémentaires.

L'opération se fait même automatiquement et l'instruction g MEM vous permet à tout moment de savoir où vous en êtes. Mais comme il n'y a en tout et pour tout 21 registres de données, ça ne peut vous mener au mieux qu'à 203 lignes de programme (1 mémoire = 7 lignes). Et dans ce cas il ne reste qu'un seul registre de données. Si l'on se place dans cette situation, on doit être passé maître dans l'art de jongler avec la pile opérationnelle pour écrire un programme utile à quelque chose.

Dans la bonne tradition

Que peut-on mettre dans ces pas ? Commençons par les étiquettes. Il y a 5 étiquettes alphanumériques qui sont toutes appelées par le préfixe f, ce qui n'en facilite pas l'emploi. Toutefois, en mode *user*, ces touches d'utilisateur sont appelées directement : c'est tout de même beaucoup plus pratique, mais ce n'est pas très cohérent et les personnes qui sont un peu distraites de temps à autre auront l'occasion de s'en apercevoir. Si l'on a oublié de vérifier l'indicateur *user* au bas de l'affichage, on obtient un message d'erreur ou le lancement d'un programme alors que l'on demandait la racine carrée d'un nombre...

La machine dispose également de 9 étiquettes numériques qu'on utilisera pour l'adressage relatif par GTO ou l'étiquetage des sous-programmes. Rien à redire à cet adressage relatif qui simplifie beaucoup la mise au point des programmes. Quant à l'adressage absolu, il est également possible.

Les commandes SST et BST permettent de lister le programme avec — et c'est nouveau chez HP — l'avance automatique sous pression continue. C'est tellement pratique que l'on regrette que ce dispositif n'ait pas été prévu sur la 41C. L'insertion est automatique et la suppression se fait avec la touche " flèche en arrière ". Il n'y a pas de NOP, et c'est logique puisque cette instruction est inutile en adressage relatif.

La batterie de tests est une classique du genre : elle est très complète ; on ne voit pas ce que l'on pourrait trouver à y redire. Les compteurs de boucle travaillent sur le registre d'index (le même que la HP 67) avec le même format que la HP 41C (format complexe contenant la valeur instantanée de l'index, la valeur finale et le pas).

Remarquons que le registre d'index permet l'adressage indirect des mémoires et qu'il autorise également un adressage indirect absolu sur un numéro de lignes, un adressage relatif et même un appel indirect de sous-programmes. Autant de possibilités qui feront peut-être regretter les limites de la mémoire disponible pour les programmes.

Un mot sur la notice. L'exemplaire que j'ai eu entre les mains était dactylographié, ce n'est donc pas la version définitive. Mais si elle n'est pas sévèrement remaniée in extremis, je peux prédire que l'épreuve finale ressemblera beaucoup aux autres notices du même constructeur : épaisse certes, mais passablement " fouillis " ; il faudra sans cesse se reporter à des annexes pour obtenir des explications détaillées.

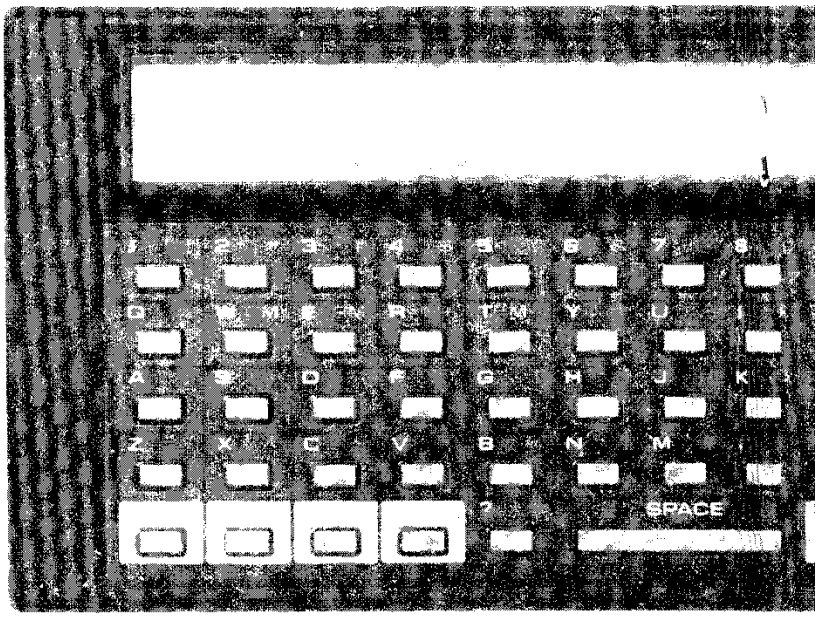
Que conclure de tout cela ? Incontestablement, la HP 11C est une honnête machine de milieu de gamme. Avec l'évolution rapide de la technologie, je m'étais habitué à être périodiquement très agréablement surpris. Peut-être suis-je légèrement blasé. Il y a seulement deux ans, devant une machine comme celle-ci, je me serais émerveillé. Aujourd'hui, mon attente est un peu déçue. D'autre part, je ne connais pas encore le prix de la machine, mais j'ai entendu parler de 1 100 FF, et si cela devait se confirmer, je me dirai que le " milieu de gamme " s'éloigne singulièrement du " bas de gamme ".

Il est vrai que la 11C (comme la 12C) disposent de tests qui vérifient le bon fonctionnement du clavier et du calcul et que c'est une machine très robuste : la qualité se paye.

☐ Xavier de La Tullaye

(2) On pourrait présenter les choses de façon plus alléchante en parlant de 203 pas de programme (la machine peut effectivement emmagasiner des programmes allant jusqu'à 203 pas), mais il faudrait ajouter alors qu'elle n'a plus qu'une seule mémoire de données, ce qui est beaucoup moins alléchant ! Nous reviendrons sur ce point.

Le premier ordinateur de poche av
et périphériques évolués,
Le premier " BASIC de poche
avec son processeur spécialisé
Et voici un nouveau venu, av
et un microprocesse



Le H
Matsushita

■ Matsushita, dont dépendent les marques Quasar, National Panasonic, etc., est un groupe japonais qui emploie quelque cent cinquante mille personnes. En collaboration avec la société franco-américaine Friends Amis, ce groupe vient de réaliser un ordinateur de poche : le Hand Held Computer.

Physiquement, le HHC occupe un volume de 22,7 × 9,5 × 3 cm et il pèse environ 500 grammes. C'est donc une machine vraiment portative même si elle est sensiblement plus encombrante que le PC 1211. Elle occupe *grosso modo* le volume d'un livre de poche. Je parle ici, bien entendu, de l'unité centrale, de l'ordinateur proprement dit qui est autonome. Nous verrons plus loin ce qu'il en est des périphériques auxquels il peut se connecter. Le boîtier a été réalisé dans une matière plastique qui semble assez robuste et il est intérieurement doublé par un blindage métallique destiné à protéger les circuits de la machine contre les champs électriques extérieurs.

Côté face, un afficheur à cristaux liquides et le clavier. Côté pile, une petite trappe donne accès à quatre cases dont l'une est déjà occupée par une " puce " et un interrupteur général permettant de réinitialiser l'ordinateur (photo 2).

Sur la gauche de l'appareil, une ouverture astucieusement refermée par un petit volet à ressort donne accès à un connecteur *mâle :* c'est là que viendront se brancher les éventuels périphériques (imprimante par exemple) ou les cartouches d'extension de mémoire. Ce collecteur mâle est tout bêtement constitué par l'extrémité de la carte imprimée où sont réunis les composants de l'ordinateur. J'espère qu'il est à toute épreuve : ce serait vraiment rageant d'avoir à changer d'ordinateur uniquement parce que l'on a cassé une prise...

La conception modulaire du système conduira en effet le plus souvent à des branchement et à des débranchements très fréquents sur ce connecteur. Sera-t-il garanti pour 1 000, 2 000, 10 000 de ces branchements ?

Affleurant sur la droite, une petite pièce métallique percée semble destinée à servir de point d'attache à une dragonne : le Hand Held Computer comme son nom l'indique en anglais est un ordinateur " à main ". Rien à signaler sur le haut et le bas du boîtier.

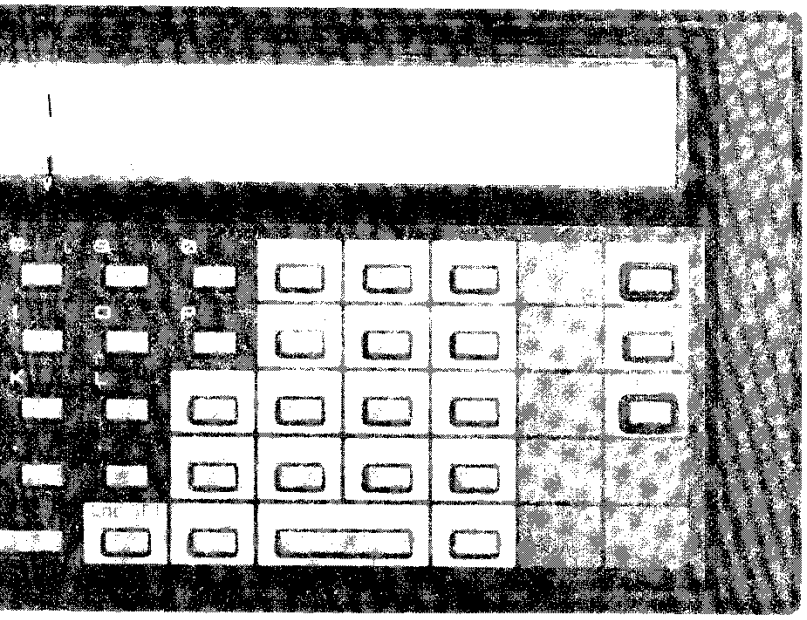
Il faut voir l'affichage

Nous avons maintenant fait le tour et nous pouvons revenir plus en détail sur le côté face. L'affichage n'est pas grand — format de poche oblige — mais c'est pour l'instant un des plus beaux du genre : cristaux liquides noirs sur fond gris, dans une matrice de 8 points sur 159, protégé par un " double vitrage ".

Le jeu des caractères disponibles comprend les majuscules comme les minuscules, et il y a même beaucoup de caractères accentués. Je reconnais des accentuations utilisées en espagnol, en allemand et dans les langues nordiques, et il y a presque tout ce qui est utile en français mais je n'ai tout de même pas trouvé la cédille : tant pis pour le français. En revanche, je remarque un alphabet japonais !

Le clavier est agréable, et sur le modèle que j'ai essayé, je n'ai

che avec langage alphanumérique
olués, c'était la HP 41C.
poche ", c'était le PC 1211
cialisé et son BASIC restreint.
enu, avec un " vrai " BASIC
ocesseur classique.



HHC
ita / Quasar

observé aucun phénomène de rebond. Les touches, 65 au total, sont petites et relativement espacées les unes des autres : le risque d'erreur de frappe en est réduit d'autant.

Que
voulez
vous ?

La partie gauche est pour l'essentiel composée d'un clavier alphanumérique complet disposé en QWERTY, la rangée supérieure étant celle des 10 chiffres. Il n'y a pas, comme sur d'autres machines, un pavé numérique distinct facilitant l'introduction rapide des données chiffrées dans des applications de calcul.

La droite du clavier comporte 21 touches de fonction dont 3 sont préprogrammables. Ces touches permettent le déplacement du curseur dans les quatre directions ; l'insertion, la destruction et la recherche d'une chaîne de caractères dans la mémoire ; la demande de la signification d'une touche (HELP), le réglage de la vitesse d'affichage, etc. Il y a même deux touches " Shift ". En résumé, beaucoup de possibilités intéressantes dont certaines sont inattendues sur une machine de poche.

A sa mise en route, le système affiche succesivement, en anglais, quatre options possibles qui correspondent à quatre programmes différents :
. 1 calculatrice
. 2 agenda
. 3 répertoire de données
. 4 BASIC Microsoft.

Tant que l'on a pas choisi entre ces quatre options, elles sont de nouveau proposées (affichage en rouleau), avec une vitesse de défilement que l'on peut d'ailleurs faire varier en utilisant la touche STP/SPD. On choisit en indiquant le numéro retenu au menu.

L'option n°1 (calculatrice) transforme l'ordinateur en une calculatrice vraiment banale et ce n'est certainement pas ce qui fera l'intérêt du HHC.

L'option n°2, sans doute plus utile, permet de programmer le micropoche à la façon d'une montre-réveil très sophistiquée : on indique un jour, une heure et un texte que la machine enregistre et qu'elle vous restituera le moment venu. Vous pouvez ainsi prévoir que votre ordinateur sonne à minuit, le 31 décembre pour vous afficher un message de bonne année. Ce " pense-bête " sonore et visuel rendra certainement de grands services à tous ceux qui, comme moi, utilisent les mouchoirs principalement pour y faire des nœuds. Plaisanterie mise à part, cette fonction utilitaire de la machine n'est pas à dédaigner puisqu'on peut enregistrer, consulter, modifier, supprimer, une grande quantité de rappels à l'ordre que la machine ne manquera pas d'exécuter le moment venu, à condition toutefois que l'on n'ait pas oublié de l'avoir avec soi !

L'option n°3 transforme la machine en un répertoire de messages alphanumériques. Vous pouvez vous en servir par exemple pour inscrire les adresses et les numéros de téléphone de vos relations, le protocole en vigueur au Saint-Siège ou n'importe quoi. Ce bloc-notes électronique est bien organisé ; on peut y effectuer des recherches très rapides, c'est une véritable petite banque de données.

Il y a aussi le BASIC

Ces deux dernières options seront certainement très appréciées par beaucoup d'utilisateurs. La quatrième (programmation en BASIC) ne présentera peut-être que peu d'intérêt pour la plupart des utilisateurs visés par ce micropoche. Mais aux yeux de ceux pour qui on ordinateur est une machine que l'on peut soi-même programmer, c'est bien entendu la plus importante, et de loin !

Le langage de programmation est le BASIC 8 kilo-octets de la société Microsoft. A peu de chose près, il est identique à celui qui équipait les premiers Apple. Notons au passage que le microprocesseur qui est au centre du système est un 6502, celui-là même dont sont pourvus l'Apple et le PET. Bien que sa consommation électrique ait été considérablement réduite (les batteries de l'unité centrale assureraient une autonomie d'une centaine d'heures), la rapidité d'exécution des programmes en BASIC est au moins égale à celle que l'on observe sur la plupart des ordinateurs individuels " de table " ! L'utilisateur pourra par ailleurs, au moyen des instructions PEEK et POKE, programmer en langage machine. Ce sont là, incontestablement, de sérieux atouts pour un micropoche.

Cependant, cette machine ne dispose, dans sa version de base, que de 2 kilo-octets de Mémoire Vive MEV, extensibles à 4 par l'enfichage dans le boîtier d'une capsule spéciale. Or, c'est dans cette même MEV que sont inscrites à la fois les informations utiles à l'utilisation de l'agenda et de la banque de données : un programmeur en verra très vite le bout. Bien sûr il existe des modules enfichables externes à l'unité centrale, mais il reste que, dans sa version de poche, cet ordinateur ne permettra d'utiliser que des programmes en BASIC assez courts. Je n'ai par ailleurs pas trouvé comment sauvegarder des programmes sur cassettes et j'ai seulement entendu parler d'un projet de " microdisquette ".

Le petit boîtier de l'ordinateur que j'ai examiné pouvait contenir plus de 60 Ko de mémoire morte MEM : la machine semble surtout destinée à l'utilisation de modules de MEM enfichables (programmes ou données) et elle correspond donc à un visage de l'informatique de poche bien spécifique.

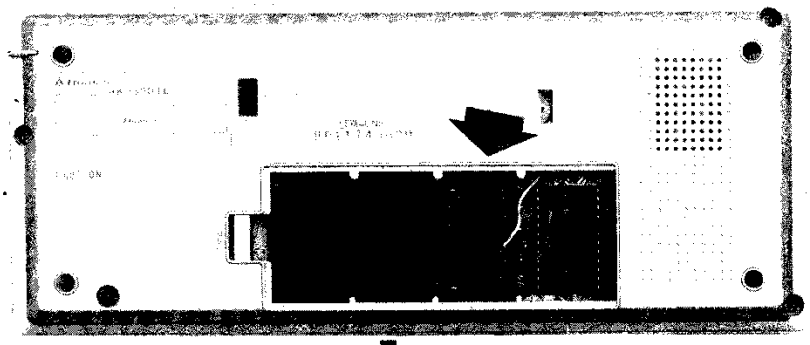


Photo 2 : la trappe soulevée laisse voir 4 emplacements disponibles pour des mémoires enfichables. Ici (flèche), l'interprêteur BASIC.

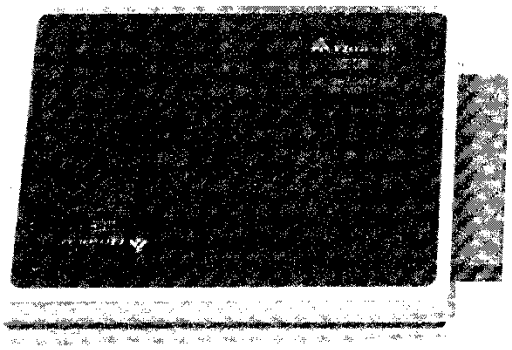


Photo 3 : le module de mémoire MEV se connecte par la prise de droite.

Toute une panoplie d'extensions

L'unité centrale peut fonctionner toute seule, mais elle est aussi le cœur d'un système de conception modulaire : c'est ici qu'intervient le connecteur mâle mentionné plus haut. Par l'intermédiaire de ce connecteur, on peut ajouter au boîtier central différentes extensions.

Une sorte de " prise multiple " autorise six branchements simultanés. Les autres extensions sont :

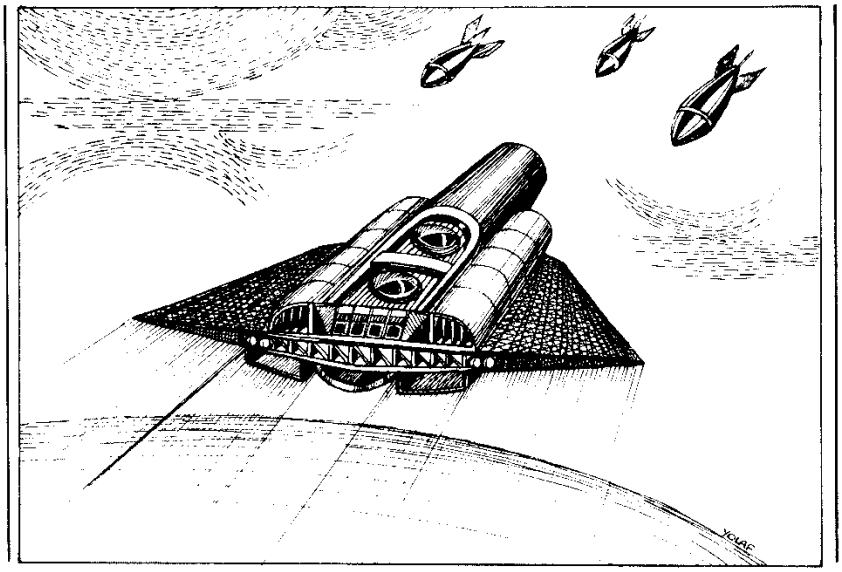
. 1 : des modules de mémoire vive MEV permanente contenant *jusqu'à* 8 Ko chacun (photo 3).
.2 : des modules de mémoire morte MEM contenant *jusqu'à* 64 Ko.
. 3 : un modem acoustique pour la communication des programmes et des données via le téléphone.
. 4 : une interface vidéo complétant l'affichage monoligne à cristaux liquides par l'écran d'un poste de télévision : on visualisera ainsi 16 lignes de 32 caractères, ou un " graphisme " de 48 × 64 points en noir et blanc (ou en couleurs lorsque les problèmes dus au procédé français SECAM seront résolus).
. 5 : une imprimante thermique à lignes de 16 caractères, dans une matrice de 5 × 8, fonctionnant à la vitesse de 24 car./sec.).
. 6 : une interface série RS 232C pour la connexion d'autres imprimantes, etc.

Tout cela avec le chargeur de batterie trouve sa place dans une mallette luxueuse et solide de style " attaché-case " destiné à faciliter le transport du système complet qui n'est donc alors pas vraiment de poche !

Au moment où vous lirez ces lignes, il est probable que ce produit sera déjà en vente aux USA. L'unité centrale seule s'y vendrait environ 500 dollars (3 000 FF.) et la mallette complète (mais sans module de MEM) 1 200 dollars (7 200 FF.). Nous le verrons distribué en Europe sans doute avant six mois.

Voici donc un matériel qui présente des aspects nouveaux ; son intérêt et son succès dépendront en grande partie de l'étendue de la bibliothèque des programmes " tout faits " dont il disposera sous forme de capsule de MEM.

☐ Jean Drano



Laser-oméga un jeu plein d'avenir

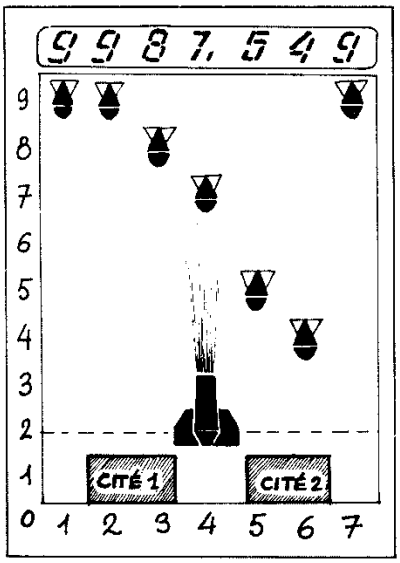
On peut faire dire beaucoup de choses à l'affichage numérique du micropoche. Ici, il vous projettera en pleine science-fiction.

■ En ce temps-là, tout était possible, même le pire, et le pire était survenu : on devrait toujours se méfier du possible ! Jamais sans doute on ne saura vraiment ce qui déclencha cette guerre, mais les rares survivants sont maintenant tous d'accord : ce fut un grand malheur de ravager la galaxie.

Le jour où la grande alerte survint dans le hurlement des sirènes, nous étions tous incrédules dans notre avant-poste aux confins de Cirius ; nous avions du mal à croire que les petits points rouges sur notre écran radar étaient des missiles et que le cauchemar avait commencé. J'avais alors le commandement du vaisseau Laser-oméga qui devait protéger les cités de Thranos et de Nygée, et c'est miracle si toutes deux n'ont pas été anéanties.

Les sept bombes de croisière dont l'écho venait d'apparaître sur notre écran n'étaient pas toutes à la même distance, mais elles suivaient des trajectoires parallèles. A bord du Laser-oméga, le mot d'ordre était " parer au plus pressé ", et tous nous l'avons appliqué du mieux que nous l'avons pu. A chaque instant, nous avions le choix entre détruire la bombe que nous avions en ligne de mire (qu'elle soit proche ou lointaine) ou nous mettre en position d'en détruire une autre si elle menaçait davantage les deux cités dont nous assurions la défense...

Fig. 1 : les bombes se déplacent de haut en bas chacune dans sa colonne et le vaisseau latéralement à l'altitude 2.



Mais l'homme est oublieux, et la vie a repris le dessus. A quoi bon vous conter mes souvenirs d'ancien combattant ? De cette bataille dans l'espace, on a fait un jeu, un petit jeu qui se dispute sur un ordinateur de poche. Et je me dis que tout est très bien ainsi : que ce cauchemar-là ne soit qu'un jeu !

Les règles du jeu

Laser-oméga se déroule sur une grille 7 × 10 qui est représentée de façon symbolique sur l'afficheur de votre Casio FX 502 P. La figure 1 illustre une des phases du jeu où l'affichage indique 9987.549. Vous devez interpréter ce nombre de la façon suivante :

. Chacun des chiffres, de gauche à

Laser-Oméga
un jeu plein d'avenir

droite, représente la distance à laquelle se trouve la bombe de croisière dans une des 7 colonnes que vous devez surveiller. Le premier chiffre de gauche (9) indique que le missile qui a surgi dans le premier secteur de gauche est encore très éloigné. En revanche, dans la deuxième colonne à partir de la droite (9987.549) le chiffre 4 vous indique que la bombe se rapproche dangereusement.

. La position du point décimal vous indique quelle est la colonne où votre vaisseau, Laser-oméga, est en position de tirer sa salve.

. Pour déplacer votre vaisseau à gauche, vous utiliserez la touche 1. La touche 3 le fera passer dans la colonne suivante sur la droite, et vous déclencherez le feu de votre batterie de laser en enfonçant la touche 2, ce qui aura pour effet de détruire le missile qui se trouve dans la colonne où vous êtes (fig. 2).

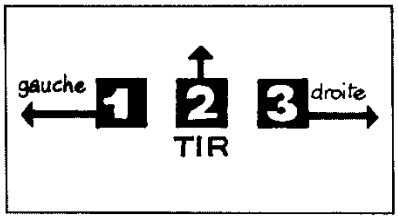


Fig. 2 : les commandes du vaisseau ont été simplifiées au possible : trois doigts suffisent.

. Il existe une autre façon de détruire un missile qui est devenu très menaçant : s'il se trouve à la distance 2 dans une des deux colonnes voisines de votre vaisseau, il suffira de le rejoindre, en enfonçant 1 ou 3, pour l'anéantir. N'hésitez pas à utiliser in extremis cette stratégie : votre Laser-oméga n'en souffrira pas.

Mode d'emploi

1. Début de partie : entrer un chiffre plus grand que zéro et enclencher Po.

2. Le nombre décimal à l'affichage vous indique la position des missiles.

3. Vous appuyez sur 1 (gauche) ou 3 (droite) pour vous déplacer, ou sur 2 pour détruire le missile qui s'approche dans la colonne où vous vous trouvez. Il est inutile d'appuyer ensuite sur EXE.

4. Si vous venez de détruire un missile, votre nouveau score apparaît.

5. L'affichage vous indique momentanément la situation que vous venez de créer et vous retournez en 2.

6. Quand une cité a été touchée, 88.8 apparaît à l'affichage suivi par le nombre total des missiles qui ont déjà atteint leur objectif.

7. Pour poursuivre le jeu, faire Mode 1 puis Po (on s'y habitue vite).

Remarque : il est important de conserver les deux "AC" dans P4 pour obtenir l'affichage de E^{010} en fin de partie.

Le jeu consiste bien entendu à protéger du mieux que l'on peut les deux cités de Thranos et de Nygée, et il s'achève lorsque dix bombes de croisière les ont atteintes ; l'affichage montre alors E^{010}. Mais pour vous soutenir le moral pendant chaque partie, le nombre des missiles que vous abattez est soigneusement décompté.

□ Eric Martin

```
PO        Mr 8      x = F     Ran       P1        10 x
x = 0     Min F     GOTO 7    x         7         =
GOTO 0    1         1         8         Min 0     P3
MAC       x = F     Ind       =         9         Min F
1         GOTO 5    M - 0     x ≥ F     LBL 0     LBL 0
Min 8     9         Ind       GOTO 9    Ind       x = F
GSB P1    x = F     Mr 0      8         Min 0     GOTO 0
LBL 0     GOTO 5    x = 0     Ind       DSZ       P4
GSB P2    Ind       GOTO 8    Min 0     GOTO 0    1
GSB P3    Min 8     Min F     GOTO 9    P2        M + 10
Min 0     1         1         LBL 8     7         Mr 10
Mr 8      M + 9     x = F     Mr 0      Min 0     Min F
Min F     Mr 9      GOTO 9    Min F     AC        9
Ind       PAUSE     3         2         LBL 0     x ≥ F
GOTO 0    LBL 5     x↔MF      x = F     ÷         GOTO 0
LBL 1     2         x ≥ F     GSB P4    1         AC
1         Min F     GOTO 9    3         0         AC
x = F     Ind       Mr 0      x = F     +         yx
GOTO 5    Mr 8      Min F     GSB P4    Ind       LBL 0
M - 8     x = F     Mr 8      5         Mr 0      8
GOTO 5    GOTO 2    x = F     x = F     =         8
LBL 3     GSB P2    GSB P5    GSB P4    DSZ       .
7         PAUSE     LBL 9     6         GOTO 0    8
x = F     7         DSZ       x = F     x         PAUSE
GOTO 5    Min 0     GOTO 6    GSB P4    (         AC 9
1         LBL 6     GOTO 0    9         Mr 8      Ind
M + 8     9         LBL 7     Ind       -         Min 0
GOTO 5    Min F     2         Min 0     1         Mr 10
LBL 2     Ind       Min F     GOTO 9    )         HLT
Ind       Mr 0
```

Pour mieux choisir "votre" ordinateur et pour mieux l'utiliser.

Lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez :

L'actualité et les tendances de l'informatique individuelle • des galops et des bancs d'essai des principaux matériels • des panoramas et des tests comparatifs • le point des grandes manifestations internationales • des articles d'initiation • des synthèses • des programmes • des interviews "exemplaires" • des conseils • des idées • des astuces.

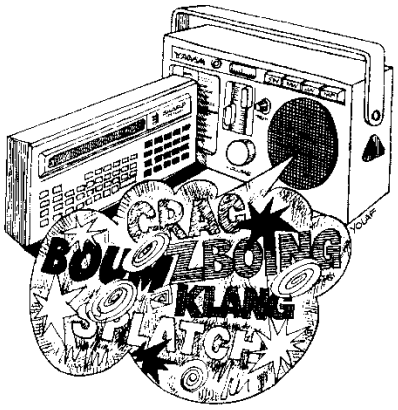
L'ORDINATEUR INDIVIDUEL

chez votre marchand de journaux

41 rue de la Grange aux Belles - 75010 Paris

Un programme qui pourrait faire du bruit...

Mettez-vous sur la bonne longueur d'ondes et écoutez battre le cœur de votre ordinateur de poche. Il a une petite musique bien à lui.



■ Chaque utilisateur du PC 1211/TRS 80 Pocket connaît l'instruction BEEP, mais il ne sait peut-être pas que sa machine est également capable de quantité d'autres effets sonores. Elle se distingue en cela de beaucoup d'autres ordinateurs de poche dont le ramage est moins agréable. Je suis par exemple parvenu à imiter la trompette de cavalerie au moment de la charge du village apache grâce à mon poquette et à un simple récepteur radiophonique.

L'ordinateur doit être déconnecté de son interface et le poste à transistor réglé sur les grandes ondes. Les petites ondes conviennent aussi, mais le résultat est moins mélodieux. Placez d'abord le coin où se trouvent les piles de la machine (à droite de l'afficheur, sous la gravure " Sharp " ou l'étiquette " TRS 80 Pocket ") près du barreau de ferrite situé à l'intérieur du récepteur : inutile d'ouvrir votre radio pour cela ! Vous pouvez maintenant rechercher la bonne longueur d'ondes. Passez France-Inter, Europe n°1, B.B.C., Radio Monte-Carlo puis R.T.L. et arrêtez-vous peu après. Mettez alors votre ordinateur en marche ; vous devez entendre votre station personnelle Radio-xxxpoche.

Si ce n'est pas le cas, modifiez légèrement la fréquence sur laquelle est votre récepteur (la bonne bande se situant aux environs de 260-285 Khz) et déplacez légèrement votre ordinateur : il n'émet que dans un rayon de quelques centimètres... Lorsque tout est au point, vous entendez très distinctement le toc toc du cœur de votre ordinateur qui attend vos instructions. Une véritable auscultation...

La touche ON produit un bruit prolongé qui rappelle celui des flippers. Les touches ▷, ◁, ↑, ↓, MODE et CL ont également chacune leur sonorité particulière d'ailleurs variable pour les quatre premières selon que l'on maintient ou non la touche enfoncée.

De plus, si votre récepteur radio est bien réglé, vous obtenez un bip à chaque pression de touche. Comme vous le verrez, la palette sonore est assez riche, et l'on y trouve aussi bien des effets musicaux que des bruitages. Les touches de déplacement du curseur, quand on les utilise sur une longue distance à l'affichage, sont des gâchettes de mitraillettes...

On peut même essayer de composer des " partitions " (musique très moderne !) qu'on exécute en mode PRO par pression des touches CL et ↑ après avoir rentré un programme.

Les programmes les plus intéressants pour l'oreille ne respectent évidemment pas la syntaxe du BASIC et l'on peut fort bien trouver des lignes telles que

1 : F = FFFFFFFF * FFFFFFFF etc.
2 : .
3 : .
4 : F = FFFFFFFF * FFFFFFFF etc.

qui ne manquent pas de rythme.

Pour terminer, je vous cite l'effet " trompette " qui me paraît assez réussi si l'on a un peu d'imagination. En mode PRO, tapez rapidement un programme du genre de celui qui est présenté ci-contre. Le plus important est de faire attention aux numéros des lignes et au nombre d'instructions : une seule ou beaucoup, ou plus ou moins en corrigeant ensuite selon l'effet obtenu. Faites CL, puis maintenez enfoncée la touche ↑ . C'est assez étonnant, n'est-ce pas ?

□ Antoine Vesigot

Inutile d'essayer de faire exécuter ce programme : Il ne "marche" pas, mais il peut tout de même se faire entendre.

```
1:F=FFFFFFF*FF
  FFFFFF*FFFFF
  FFF*FFFFFFFF
  *FFFFFFFF*FF
  FFFFFF*FFFFF
  FFF*FFFFFFFF
  *FFF
2:F=FFFFFFF*FF
  FFFFFF*FFFFF
  FFF*FFFFFFFF
  *FFFFFFFF*FF
  FFFFFF*FFFFF
  FFF*FFFFFFFF
  *FFFFF
3:GOTO 100
4:.
5:FFFFFFFFF*FF
  FFFFFF*FFFFF
  FFF*FFFFFFFF
  *FFFFFFFF*FF
  FFFFFF*FFFFF
  FFF*FFFFFFFF
  *FFFFF
6:.
7:.
8:.
9:.
10:.
11:.
12:.
13:.
42:.
50:.
51:.
52:.
53:.
54:.
85:.
100:K=LKJJJ-JKIU
    JU*OIUJU+POI
    UH-JUHYT*JUH
    Y*JUHYTGT*JU
    HYTG*IKJUHUJ
    *JU
321:H=1232564888
    7887777458:K
    =12236258947
500:MJJUHUYHYHGT
    TTTTTTTFRFRD
    EDDEDER
510:END
555:FFFFFFFFF*FF
    FFFFFF*FFFFF
    FFF*FFFFFFFF
    *FFFFFFFF*FF
    FFFFFF*FFFFF
    FFF*FFFFFFFF
    *FFF
621:.
853:.
855:.
900:.
999:LIST 999
```

Une petite DCA

Il n'est pas commun de s'entraîner en chambre au tir instinctif avec un bombardier pour cible mobile. Heureusement, il s'agit d'un faux canon !

■ Avec une TI 57 soigneusement garnie des 47 pas du programme ci-joint, on s'exerce *sans danger* à des calculs d'angles qui interviennent dans les tirs de défense antiaérienne.

Lorsqu'un bombardier lourd de menaces se dirige sur vous, il faut faire très vite, et quand on ignore en plus son attitude exacte, on est conduit à tirer un peu à l'aveuglette : un coup trop haut, un coup trop bas... Mais la cible s'est encore un peu rapprochée de vous, et les calculs de trigonométrie se font de façon intuitive !

Sur votre TI 57, les choses se dérouleront de la façon suivante : après avoir enfoncé la touche RST, vous inscrivez à l'affichage un nombre quelconque et vous pressez la touche R/S. Ce nombre permet grâce à un générateur de nombres au hasard de déterminer l'altitude du bombardier, altitude que vous ignorez toute la partie durant et qui est stockée en mémoire 1 au pas 9. Pour la calculer, le programme prend la partie décimale de la racine carrée de votre nombre et la multiplie par 200.

Cette altitude sera donc comprise entre 0 et 200 mètres et elle ne variera pas jusqu'à la fin du jeu. Au pas 6, on profite de l'affichage de 200 pour initialiser le registre 0 qui contient la distance au sol séparant votre batterie de D.C.A. du bombardier (voir la figure). La cadence de feu est telle que cette distance diminue de vingt mètres à chaque tir.

Vous devez régler la hausse du canon, c'est-à-dire indiquer l'angle de tir par rapport au sol, entre 0 et 90°. Une fois que vous avez inscrit cet angle à l'affichage, vous pressez la touche R/S et la machine vous indique en retour

. si vous avez visé trop haut : affichage de 1 (instruction pause du pas 41),

. si vous avez visé trop bas : affichage de −1 (instruction pause du pas 45),

. si votre obus de D.C.A. a exposé suffisamment près du bombardier pour le détruire : un affichage clignotant vous indique quel était le meilleur angle de tir qui est inscrit dans la mémoire 7 après avoir été calculé aux pas 14 à 19.

Après votre premier essai — qui a toute chance d'être infructueux — la machine affiche la distance au sol qui maintenant vous sépare du bombardier, et vous rectifiez le tir en entrant un nouvel angle et en pressant R/S.

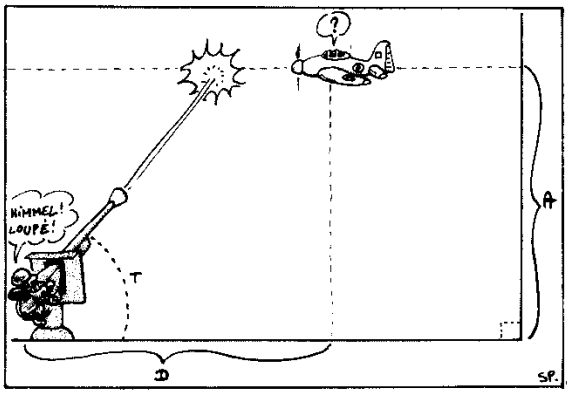
Vous avez ainsi droit à dix essais. Il est en effet inutile d'essayer de tirer à la verticale (90° de hausse) quand l'avion est au-dessus de vous : c'est trop tard, puisqu'il s'agit d'un bombardier... N'oubliez pas non plus que l'avion volant à altitude constante, l'angle correct augmente à mesure qu'il s'approche de vous.

La D.C.A. qui nous intéresse n'est fort heureusement qu'un jeu ; il n'a donc pas été tenu compte des lois de la balistique, et la trajectoire de vos obus et strictement rectiligne, ce qui rendra jaloux tous les véritables artilleurs.

☐ Arnaud Beauregard

Contenu des mémoires
M0 : distance au sol
M1 : altitude de l'avion
M2 : angle de tir
M7 : (registre de test) angle réel, ou approché, du bon tir.

```
     TIR A LA D.C.A.
 AUTEUR : ARNAUD BEAUREGARD
 COPYRIGHT L'ORDINATEUR
     DE POCHE ET L'AUTEUR
**************************
  0   24          RACINE CARREE DE X
  1   -49         INV 2ND INT
  2   55          MULTIPLIE PAR
  3   02          2
  4   00          0
  5   00          0
  6   32  0       STO  0
  7   85          =
  8   49          2ND INT
  9   32  1       STO  1
 10   86  0       2ND LBL  0
 11   33  0       RCL  0
 12   81          R/S
 13   32  2       STO  2
 14   33  1       RCL  1
 15   45          DIVISE PAR
 16   33  0       RCL  0
 17   85          =
 18   -20         INV 2ND TAN
 19   49          2ND INT
 20   32  7       STO  7
 21   02          2
 22   00          0
 23   -34  0      INV SUM  0
 24   02          2
 25   34  7       SUM  7
 26   33  2       RCL  2
 27   76          2ND X>=T
 28   51  2       GTO  2
 29   03          3
 30   -34  7      INV SUM  7
 31   33  2       RCL  2
 32   -76         INV 2ND X >= T ?
 33   51  1       GTO  1
 34   01          1
 35   34  7       SUM  7
 36   33  7       RCL  7
 37   51  9       GTO  9
 38   86  1       2ND LBL  1
 39   01          1
 40   84          +/-
 41   36          2ND PAUSE
 42   51  0       GTO  0
 43   86  2       2ND LBL  2
 44   01          1
 45   36          2ND PAUSE
 46   51  0       GTO  0
```



A : altitude de l'avion
D : distance au sol
T : angle de tir
L'angle exact est égal à l'Arc tangente de A/D.

Kibur ou le désordre alphabétique

Si votre ordinateur de poche peut afficher des lettres, le jeu proposé ici devrait vous passionner par ses nombreux renversements.

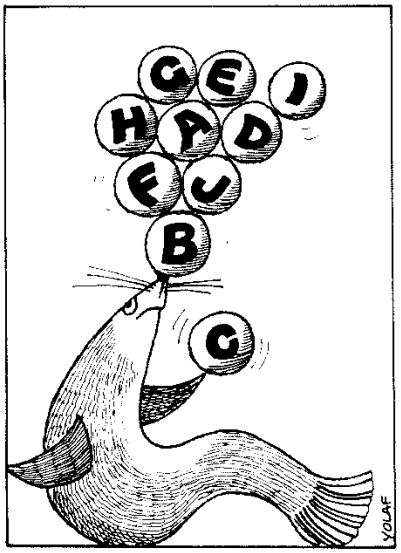
■ Le cube de Rubik est déjà un grand classique parmi les jeux, comme en témoigne la vaste épidémie de Rubikite aiguë qui sévit depuis quelque temps dans une large partie du monde. Les utilisateurs de micropoches ont peut-être été effleurés par l'idée d'adapter ce jeu sur leur machine, mais ils y ont vite renoncé, car les moyens sont manifestement insuffisants. Ce que l'on peut faire, cependant, c'est adapter un jeu du même genre, basé sur le principe du cube de Rubik.

Le programme décrit ici permet précisément de jouer à un tel jeu, appelé " jeu de Kibur " (l'origine du nom ne devrait plus avoir de mystère à présent), mais qui évidemment est loin d'avoir la complexité du cube original. Il n'en offre pas moins un intérêt certain, surtout si on y joue à plusieurs.

Le micropoche, ici une HP-41C, présente dix lettres (A à J) dans un ordre arbitraire : le but du jeu est de retrouver l'ordre alphabétique normal en un minimum de coups. Chaque coup est défini par l'inversion de deux lettres, l'une fixe, étant la première à gauche, en position " 0 ", l'autre étant au choix du joueur parmi toutes les dix lettres (de 0 à 9 : il suffit d'introduire le rang de la lettre voulue). S'il n'y avait que cela, ce serait bien trop facile et sans grand intérêt, c'est pourquoi un " effet secondaire " a été ajouté à cette inversion : c'est l'inversion des deux lettres situées de part et d'autre de celle choisie pour être inversée avec la lettre en position zéro.

Par exemple, soit la combinaison : " BFJHACDIGE ". En choisissant " 2 " (position de la lettre J), pour l'insertion on a : JHBFACDIGE ". C'est-à-dire que " B " (position 0) a été inversée avec " J " (position 2, choisie), mais en plus, les lettres " F " et " H ", de part et d'autre de " J " se retrouvent également inversées.

Signalons encore que cette ligne de lettres est bouclée, c'est-à-dire que si on joue la position " 0 ", cela a comme effet d'intervertir les lettres en position 1 et 9, tandis que



jouer " 9 " résulte en ce que la lettre " 0 " passe en " 9 ", la lettre " 9 " passe en " 8 ", et que la lettre " 8 " passe en " 0 " ! Enfin, de façon similaire, jouer " 1 " revient à faire une permutation circulaire des trois lettres en positions " 0 ", " 1 " et " 2 ". Ainsi, reprenant l'exemple précédent où nous l'avons laissé, si nous jouons " 1 " nous obtiendrons : " BJHFACDIGE " (le B a pris la place du J qui a lui-même pris la place du H, qui, à son tour, est venu occuper la position du B).

Ces particularités compliquent certainement le jeu, mais elles contribuent aussi, par là, à son intérêt, en rendant parfois la solution beaucoup moins évidente qu'il y paraît.

Le programme tel qu'il est présenté offre trois possibilités d'entrées différentes. La première, en faisant XEQ " KIBUR ", demande à la calculatrice de générer elle-même une combinaison de départ, ce qu'elle fait à partir de l'ordre alphabétique normal, en appliquant dix opérations successives aléatoires (mais identiques aux opérations de résolution) : c'est-à-dire que, théoriquement, en moyenne, on devrait pouvoir résoudre tous les problèmes posés par la calculatrice en une dizaine de coups. Il " suffit " de refaire les

Exemple de partie n° 1

joué	résultat	
	A B E D G F H J I C	: 0
8	I B E D G F H C A J	: 1
6	H B E D G C I F A J	: 2
7	F B E D G C A H I J	: 3
5	C B E D A F G H I J	: 4
3	D B A C E F G H I J	: 5
1	A D B C E F G H I J	: 6
2	B C A D E F G H I J	: 7
1	A B C D E F G H I J	: 8

Exemple de partie n° 2

joué	résultat	
	C B I F E D G A H J	: 0
2	I F C B E D G A H J	: 1
6	G F C B E A I D H J	: 2
7	D F C B E A H G I J	: 3
6	H F C B E G D A I J	: 4
5	G F C B D H E A I J	: 5
6	E F C B D A G H I J	: 6
1	C E F B D A G H I J	: 7
2	F B C E D A G H I J	: 8
3	E B D F C A G H I J	: 9
4	C B D A E F G H I J	: 10
2	D A C B E F G H I J	: 11
1	C D A B E F G H I J	: 12
2	A B C D E F G H I J	: 13

Un petit problème : à partir de la combinaison " D C E B H F A J G I " retrouver l'ordre alphabétique normal en... quatre coups !

Solution : jouer trois, puis un puis sept et enfin, neuf...

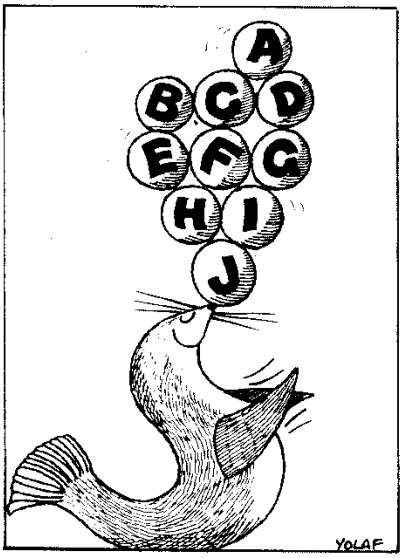
```
                PRP "KIBUR"

 01♦LBL "KIBUR"
"ABCDEFGHIJ"  SF 01
XEQ 03  10  STO 11

 07♦LBL A
RCL 00  PI  +  5  Y↑X
FRC  STO 00  10  *  INT
XEQ 02  DSE 11  GTO A
XEQ 01  ASTO 12  ASHF
ASTO 13

 25♦LBL B
FIX 0  0  STO 16

 29♦LBL C
XEQ 01  "⊢:"  ARCL 16
PROMPT  XEQ 02  99
RCL 16  1  +  STO 16
X<=Y?  GTO C  "PERDU"
PROMPT  GTO "KIBIS"

 45♦LBL "KIBIN"
"COMBINAISON?"  AON
PROMPT  AOFF

 50♦LBL 03
ASTO 12  ASHF  ASTO 13

 54♦LBL "KIBIS"
1.01  STO 11  CLA
ARCL 12  ARCL 13
ASTO 14  ASHF  ASTO 15

 63♦LBL D
"     "  ARCL 14
ARCL 15  ASTO IND 11
ASHF  ASTO 14  ASHF
ASTO 15  " "
ARCL IND 11  ASHF
ASTO IND 11  ISG 11
GTO D  FS?C 01  RTN
GTO B

 81♦LBL 01
CLA  1.01  STO 11

 85♦LBL a
ARCL IND 11  ISG 11
GTO a  RTN

 90♦LBL 02
1  +  10  X<>Y  X>Y?
X<>Y  STO 17  1  +
STO 18  RCL 01
RCL IND 17  STO 01  RDN
STO IND 17  RCL 17  1  -
X=0?  10  STO 17  11
RCL 18  X=Y?  1  STO 18
RCL IND 17  RCL IND 18
STO IND 17  RDN
STO IND 18  .END.
```

mêmes opérations, mais dans l'ordre inverse. Ce n'est toutefois pas tout à fait vrai pour les opérations " 1 " et " 9 ", qui ne sont pas réversibles, mais cycliques de période 3 : donc si la calculatrice a effectuée une de ces deux opérations une fois entre deux autres opérations, l'opération inverse est obtenue en effectuant ladite opération deux fois de suite, tandis que si une de ces deux opérations a été effectuée deux fois de suite, effectuer cette opération une seule fois suffit à donner son inverse. Malheureusement, ou heureusement, selon le point de vue auquel on se place, il n'y a aucun moyen de connaître la séquence d'opérations générée par la calculatrice, à moins de connaître la séquence aléatoire (et de sortir du programme la fonction génératrice — classique — utilisée). De toutes les façons, ce n'est pas toujours obligatoirement la solution la plus course (en particulier cette séquence aléatoire peut comporter deux fois de suite la même opération, qui s'annule donc, à part les opérations " 1 " et " 9 " dont il est question plus haut).

La deuxième possibilité d'accès, au moyen de : XEQ " KIBIN ", permet au joueur, ou à son comparse, d'introduire une combinaison de son choix, par exemple une combinaison intéressante, ou particulièrement épineuse et connue comme telle.

Le troisième mode d'accès, avec : XEQ " KIBIS ", permet de recommencer une partie avec la combinaison du début de la précédente partie jouée : cette combinaison est remise automatiquement en lice par la calculatrice, et il est ainsi très facile de jouer plusieurs parties successives à partir de la même combinaison, ce qui est particulièrement intéressant lorsqu'on joue à plusieurs, le problème étant alors identique pour tous les concurrents, chacun essayant de parvenir à l'ordre alphabétique correct en un minimum de coups.

La calculatrice affiche à la suite de chaque combinaison de dix lettres quel est le nombre de coups déjà joués. Cela permet de savoir à tout moment où l'on en est. Si plus de 99 coups ont déjà été joués, la calculatrice affiche " PERDU " et effectue automatiquement une entrée en " KIBIS ", c'est-à-dire qu'elle reprend la combinaison de départ pour une nouvelle tentative.



Lorsque l'on a fait son choix d'opération sur le clavier et que l'on a tapé une touche numérique, la situation affichée (combinaison actuelle et nombre de coups joués) disparaît, mais elle est conservée dans le registre alpha de la machine tant que l'on n'a pas appuyé sur la touche R/S ; il est donc possible, si on a un doute de toute dernière seconde, de vérifier encore une fois cette situation, mais il ne faut pas oublier de repasser en mode numérique une fois cette vérification faite, faute de quoi la machine signalera une erreur " ALPHA DATA " !

☐ Robert Pulluard

Utilisation du programme :

A : variante Kibur, la calculatrice génère elle-même une combinaison.

1. faire XEQ " KIBUR "
2. introduire le numéro de position de la lettre que l'on désire intervertir avec la première et presser sur R/S. Attention : la première lettre est en position zéro, la suivante en 1, puis 2, etc.
3. retourner en 2.

B : variante KIBIN, le joueur ou l'un de ses partenaires introduit la combinaison de départ.

1. faire XEQ " KIBIN ". L'affichage indique " COMBINAISON ", et vous introduisez les dix premières lettres de l'alphabet dans l'ordre que vous voulez. Une pression sur R/S et l'on se retrouve à l'étape 2 de Kibur.

C : variante de KIBIS, la calculatrice reprend la dernière combinaison de départ utilisée (ce qui permet de recommencer la même partie en essayant d'améliorer son score). Faire XEQ " KIBIS ", puis procéder comme à l'étape 2 de Kibur.

TEXAS-INSTRUMENTS TI 58C

● *10 chiffres rouges* ● *Affichage LED* ● *60 mémoires ou 480 pas de programmes permanents* ● *Module interchangeable préprogrammé de 5 000 pas environ contenant 25 programmes divers* ● *Trigo* ● *Log* ● *Moyenne, Ecart-type* ● *Régression linéaire* ● *Corrélation* ● *En option : 5 modules préprogrammés et une imprimante* ● *Alimentation par batteries rechargeables et secteur* ● *Autonomie 3 heures.*

164 × 80 × 37 mm

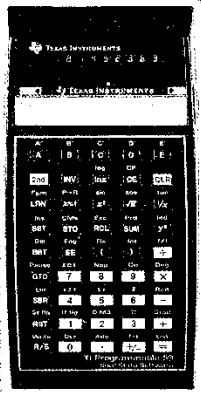
715 F ttc

Performances / Prix Très bonnes

Qualité : Bonne

Adaptable toutes professions par modules et mémoire permanente.

La Polytechnicienne



TEXAS-INSTRUMENTS TI 59

● *Mêmes caractéristiques que la TI 58, sauf les suivantes* ● *Capacité de programmation 960 pas ou 100 mémoires et 160 pas* ● *Lecteur enregistreur de cartes magnétiques pour l'enregistrement des mémoires et des programmes* ● *Possibilité d'écriture alphanumérique et de trace de courbes par points avec le bloc imprimant en option*

164 × 80 × 37 mm

1350 F ttc

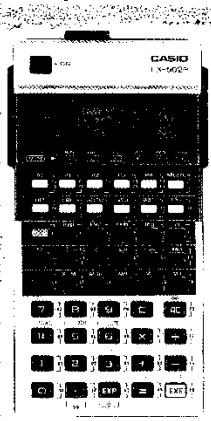
Performances / Prix Très bonnes

Qualité : Bonne

s'approche de l'ordinateur par ses possibilités.

De très loin la moins chère à cartes.

La Super-Mémoire



CASIO FX 502 P

● *10 chiffres noirs sur fond jaune* ● *Affichage LCD* ● *22 mémoires permanentes* ● *Notation prioritaire* ● *Trigo* ● *Log* ● *n !* ● *Moyenne, Ecart-type* ● *256 pas combinés de programmes permanents* ● *Option interface pour magnétophone à cassettes* ● *Autonomie jusqu'à 1 300 heures.*

144 × 75 × 15 mm

590 F ttc

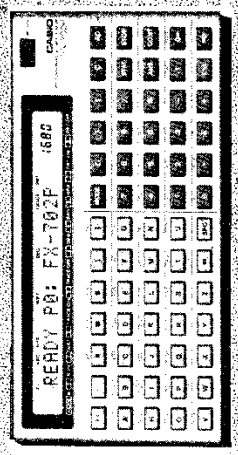
Interf. magnéto : 189 F

Performances / Prix Bonnes.

Qualité : Bonne

une programmable puissante réellement de poche.

L'ex-futur



Casio 702P

■ *Micro-ordinateur de poche* ● *Langage Basic* ● *Très grande rapidite de calcul* ● *De 1680 pas + 26 mémoires à 80 pas + 226 mémoires* ● *Nombreuses fonctions au clavier, dont Trigo, Log, Stat, régressions, corrélations.* ● *Capacité 10 chiffres* ● *Affichage 20 caractères.*

Prix : 1320 F ttc.

82 × 165 × 17 mm

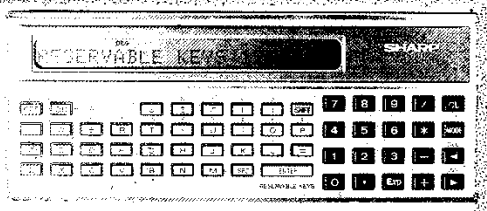
3 options ultérieures : ● *Interface magnéto* ● *Extension mémoire programmable* ● *Imprimante.*

Performance/Prix : Très bonnes.

Qualité : Bonne.

Beaucoup de fonctions au clavier avec la programmation en basic.

La Fascinante



50 Millions de Calculs *au Banc d'Essai Duriez*

IMPARTIAL, Duriez a testé toutes les calculatrices possibles : les marques établies — quelquefois surfaites ou dépassées — et les jeunes qui percent en force.

Duriez n'est pas chargé de soutenir telle ou telle marque. Il vous aide à voir clair et acheter sûr. Il comprime les prix. Si vous n'êtes pas satisfait sous 8 jours, il vous rembourse.

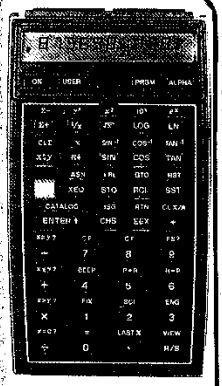
Parmi les 38 modèles du Palmarès Duriez (voir catalogue-banc d'essai, gratuit), en voici 7. Dignes de Duriez.

Duriez, fondé en 1783 (Nombre Premier), 132, Bd St-Germain, 6e, Mo et RER Odéon - St-Michel - Luxg. 9 h à 19 h du Ma. au Sa.

Prix TTC au 15.10.81

Tous modèles de cette page garantis 1 an.

Toutes bibliothèques et accessoires en stock.



HEWLETT-PACKARD 41C

● *Affichage alphanumérique noir sur fond LCD gris* ● *12 caractères alphabétiques* ● *130 fonctions préprogrammées* ● *Mémoire à 63 registres permanents de données (1 registre = 7 lignes de programme ou 1 mémoire de données)* ● *6 niveaux de sous programmes* ● *Adressage indirect sur tous les registres* ● *Configuration modulaire* ● *Nombreux logiciels et livrets d'applications* ● *Autonomie jusqu'à 1 000 heures.*

144 × 79 × 33 mm

1790 F ttc

Performance/Prix : *Bonne*

Qualité : *Très bonne*

remarquable par ses possibilités d'extensions.

La Surpuissante

Extensions de la HP 41 C :

I. Jusqu'à 4 modules de mémoires programmables supplémentaires, comportant chacun 64 registres ● *Supplément au prix de base* **260 F ttc**

II. Nombreux modules préprogrammables ● *Mathématiques* ● *Statistiques* ● *Finances, etc.* ● **260 F ttc** *sauf except*

III. Lecteur enregistreur de cartes magnétiques. Les cartes enregistrées pour le modèle HP 67 et 97 sont compatibles, ce qui permet d'utiliser les bibliothèques et fascicules de programmes existants pour ce modèle ● *Prix :* **1500 F ttc**

IV. Imprimante thermique alphanumérique permettant le tracé de courbe par points ● *Prix :* **2500 F ttc**

3 201530 041021

V. Lecteur optique de code-barres pour introduction rapide des programmes ● *Prix :* **940 F ttc**

HEWLETT-PACKARD 41CV

Mêmes caractéristiques que la 41C, sauf 319 registres.

2390 F ttc

SHARP PC 1211

● *Micro-ordinateur de poche* ● *Affichage LCD 24 caractères alphanumériques noirs sur fond jaune* ● *Capacité 10 chiffres* ● *Langage Basic* ● *1 424 pas de progr. permanents (ou 178 mémoires) + 26 mémoires indépendantes permanentes* ● *Mini clavier machine à écrire* ● *Option interface pour magnétophone* ● *Etui plastique rigide* ● *Autonomie jusqu'à 300 h.* ● *Manuels d'utilisation, de Basic, d'applications (79 programmes divers).*

71 × 177 × 17

1190 F ttc

Imprimante : 895 F ttc

Performance/Prix : *Très bonne*

Qualité : *Bonne*

idéal pour apprendre le Basic et très performante pour sa taille.

Le Basic en Poche

HEWLETT-PACKARD 34 C

● *10 chiffres rouges* ● *Affichage LED* ● *Notation polonaise inverse* ● *Mémoire permanente* ● *Mémoire dynamique - configuration de base : 70 lignes de programmes et 21 registres de données ; possibilité d'avoir jusqu'à 210 lignes de programme* ● *Deux touches à définir A et B* ● *6 niveaux de sous-programmes* ● *Adressage indirect et relatif* ● *Fonction SOLVE : calcul des racines de plusieurs types de fonction* ● *Fonction INTEGRATE : calcul de l'aire d'une courbe de fonction f(x) dans un intervalle spécifié, y compris maxima et minima.*

142 × 75 × 30 mm

1090 F ttc

Performance/Prix : *bonne*

Qualité : *Bonne*

permet la résolution directe de la plupart des calculs scientifiques.

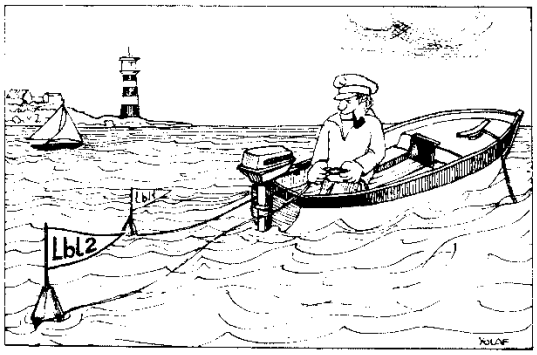
La Super-Savante

commandez /Duriez par poste

Joindre règlement à commande. Expédition sans frais. Garantie 1 an, pièces et main-d'œuvre. Satisfait ss 8 jours ou remboursé.

• Duriez 132, Bd St Germain, 6e. Mo Odéon.

Média Conseil - Neuilly



la programmation : les boucles et les tests

Quand un programme se complique un peu, il faut y poser des jalons. Ce sont les étiquettes.

■ Vous savez sans doute que pour pêcher les crabes il y a deux techniques. La première consiste à aller à marée basse entre les rochers avec un crochet : on attrape les bêtes en soulevant les pierres, il faut donc être sportif et ne pas trop compter sur cette méthode pour faire bombance. La seconde technique demande beaucoup plus de matériel : des casiers, un bateau pour aller les mouiller aux endroits propices et des bouées pour repérer où ont été immergés les casiers. (Rassurez-vous, c'est bien *l'Ordinateur de poche* que vous êtes en train de lire...).

Ecrire un programme pour une TI57 ou pour une autre machine présente de grandes similitudes avec cette pêche : ou le programme est simple et il ne demande que peu de choses ou bien il est compliqué et il faut mettre en œuvre de plus gros moyens. Jusqu'ici, nous avons réalisé des programmes simples, et la méthode suivie ressemblait à la pêche au crochet : résultats immédiats, pas de complications, mais peu d'espoir de faire des prises extraordinaires. Préparons maintenant nos casiers pour une pêche plus substantielle.

Lorsque vous mouillez une nasse, l'essentiel de la pêche va se dérouler au fond de l'eau, hors de votre vue. Cette masse est reliée à la surface par un filin où se trouve attachée une bouée de couleur vive. Cela permet de repérer le casier de loin quand le moment est venu de le relever.

En programmation, c'est à peu près la même chose. Tant que vous ne vous éloignez pas du bord (entendez le pas 00), vous n'avez pas besoin de repère, mais si vous voulez réaliser des programmes longs et performants, vous aurez besoin de bouées pour retrouver le début des différentes phases.

L'étiquette : un point de repère, une bouée

Ces bouées prennent le nom d'étiquettes (labels en anglais). Vous allez les placer à chaque endroit du programme qui aura besoin d'être repéré pour l'exécution d'une phase particulière. Quand vous voudrez récupérer le contenu du casier (la portion de programme), il vous suffira d'aller à la bouée (l'étiquette) avec votre barque (le pointeur) et de tirer (lancer l'exécution).

Puisque nous sommes dans les affaires maritimes, nous allons réaliser un programme pour convertir les kilomètres en milles nautiques et inversement. Nous placerons une étiquette au début de chaque sous-ensemble pour pouvoir récupérer celui qui nous intéresse au moment voulu.

Traçons d'abord l'organigramme correspondant. Nous utiliserons comme symbole de l'étiquette le signe qui marque le début ou la fin d'un programme et nous inscrivons " Lbl " à l'intérieur pour éviter toute confusion :

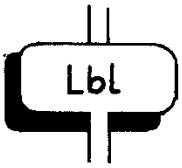


Fig. 1

Un identificateur de 0 à 9 numérotera les étiquettes. Sur la TI 57, l'identification est uniquement numérique. Sur d'autres machines, il peut être alphabétique, ou être un nom de fonction. D'autres matériels ne possèdent pas d'étiquettes. On doit dans ce cas se repérer sur les numéros de pas.

Voici un exemple de programme écrit pour TI 57 et utilisant deux étiquettes :

00	01	1
01	83	•
02	08	8
03	05	5
04	02	2
05	32 1	STO 1
06	86 1	2nd Lbl 1
07	45	÷
08	33 1	RCL 1
09	85	=
10	81	R/S
11	86 2	2nd Lbl 2
12	55	x
13	33 1	RCL 1
14	85	=
15	81	R/S

Pour exécuter le programme, faites RST puis R/S. Entrez une dis-

tance à convertir et faites GTO1, R/S si la conversion désirée est km en milles. Sinon faites GTO2, R/S après avoir rentré la valeur. Ce petit programme peut être utilisé pour convertir toute sorte de données, en remplaçant la constante du début de programme. Si par exemple, vous désirez convertir des centimètres en inches (mesure anglaise) et inversement, la constante est 2,54. Pour convertir des mètres en pieds, 0,305 ; des kilomètres en *miles* (attention, ce ne sont pas les mêmes que tout à l'heure, ici n'est la mesure anglaise) 1,609. Vous obtiendrez la conversion de francs français en dollars en utilisant 5,95. Mais attention, car ce n'est plus une constante, et il faut réajuster en fonction du cours du change...

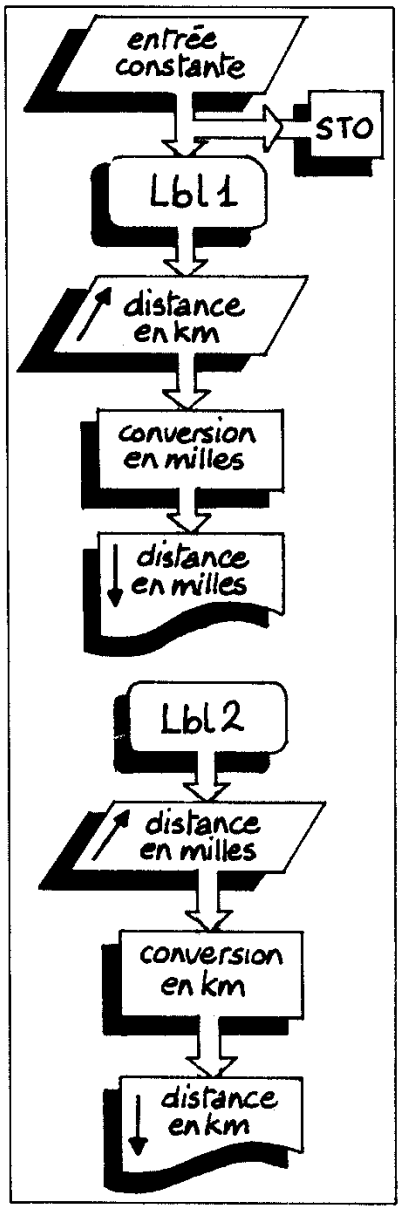


Fig. 2 - Organigramme de la conversion des kilomètres en milles et inversement.

Un ordre impératif : Va !

Nous avons lancé l'exécution des sous-ensembles du programme en utilisant la touche GTO suivie du numéro de l'étiquette que nous voulions appeler. GTO est l'abréviation de l'impératif anglais Goto qui signifie " va jusqu'à ". Cette instruction envoie le pointeur de programme sur l'étiquette dont le numéro est spécifié par l'identificateur de GTO.

Nous avons utilisé l'instruction en mode d'exécution, mais elle est également programmable. Placée à un endroit quelconque du programme, elle positionnera le pointeur à l'étiquette définie. C'est pour cela qu'on l'appelle *branchement obligé,* ou *inconditionnel.*

Cet ordre a deux utilisations principales. Il permet d'une part d'exécuter des bouclages lorsque le début de la boucle n'est pas au pas 00 (dans ce cas, nous avons déjà vu que c'est l'instruction RST qui est utilisée) et il est d'autre part employé pour sortir d'un test. Nous étudierons cette application un peu plus loin.

Si vous prenez l'habitude d'utiliser GTO en dehors de ces deux cas, c'est le signe que vous ne passez pas suffisamment de temps à étudier les algorithmes de vos programmes : ils manquent de rigueur.

Nous venons de voir que sur TI 57, l'ordre GTO renvoie nécessairement à une étiquette marquant le début d'une phase de programme. On parle dans ce cas-là d'adressage *relatif.* Ce mode est à opposer à l'adressage *absolu* qui existe sur beaucoup d'autres ordinateurs de poche (HP33, TI58-59, et dans tous les BASIC). L'identificateur de GTO est alors un numéro de pas, et le branchement se fait " à la volée ", sans qu'il n'y ait de repère spécial dans le programme.

L'adressage absolu présente l'avantage d'une plus grande rapidité de transfert, mais il complique en revanche beaucoup la mise au point des programmes puisque chaque insertion ou suppression de pas entraîne une modification des numéros de pas dans l'identificateur (ce n'est toutefois pas le cas en BASIC

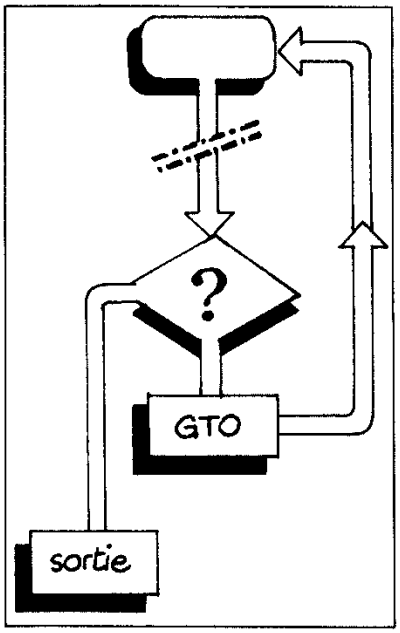


Fig. 3

puisque les numéros de ligne ne changent pas lors de l'utilisation d'une commande d'édition).

Sur la TI 57, on peut cependant utiliser l'adressage absolu en mode calcul, pour envoyer le pointeur sur un pas donné avant de lancer l'exécution. Cette possibilité rend des services pour la mise au point des programmes. Il faut faire pour cela GTO 2nd nn, nn représentant les deux chiffres du numéro de pas.

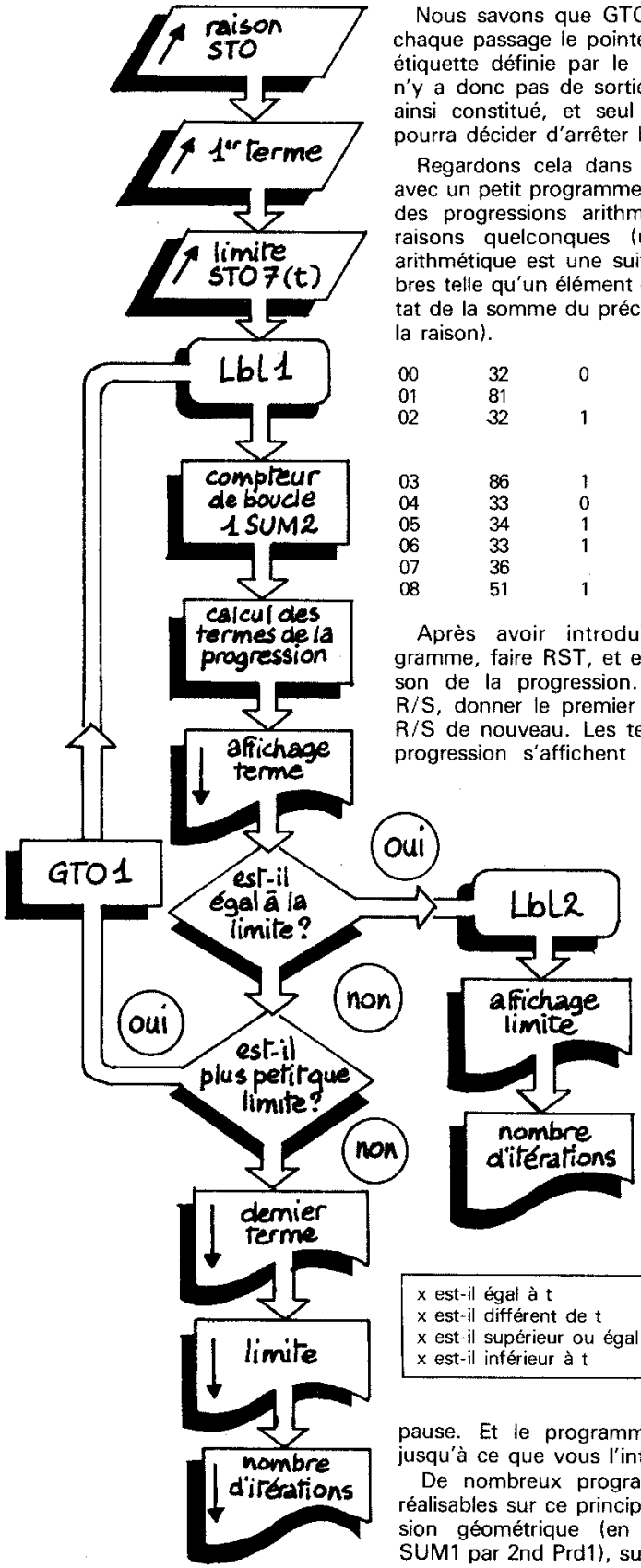


Fig. 4

Nous savons que GTO renvoie à chaque passage le pointeur sur une étiquette définie par le pointeur. Il n'y a donc pas de sortie au circuit ainsi constitué, et seul l'opérateur pourra décider d'arrêter la course.

Regardons cela dans la pratique avec un petit programme qui calcule des progressions arithmétiques de raisons quelconques (une raison arithmétique est une suite de nombres telle qu'un élément est le résultat de la somme du précédent et de la raison).

00	32	0	STO 0
01	81		R/S
02	32	1	STO 1
03	86	1	2nd Lbl 1
04	33	0	RCL 0
05	34	1	SUM 1
06	33	1	RCL 1
07	36		2nd Pause
08	51	1	GTO 1

Après avoir introduit le programme, faire RST, et entrer la raison de la progression. Après un R/S, donner le premier terme puis R/S de nouveau. Les termes de la progression s'affichent pendant la pause. Et le programme continue jusqu'à ce que vous l'interrompiez.

De nombreux programmes sont réalisables sur ce principe : progression géométrique (en remplaçant SUM1 par 2nd Prd1), suites de tous genres : de Fibonacci, de Riemann...

Nous pourrions même réétudier le " chronomètre " réalisé dans le précédent numéro à partir d'un algorithme similaire à celui que nous venons d'utiliser = mise en mémoire d'une constante, conversion décimale et addition en mémoire puis rappel et conversion sexagésimale avant l'affichage pendant une pause et enfin bouclage sur une étiquette située avant la conversion décimale.

Une sortie de secours

Tous ces programmes fonctionnent parfaitement, mais ils obligent à surveiller la calculatrice pendant tout son temps de travail jusqu'à ce qu'elle atteigne le résultat désiré. Et cela peut parfois prendre du temps. Fort heureusement, la calculatrice peut se surveiller elle-même. Et nous obtiendrons cela au moyen d'un test (voir fig. 3). Un losange a été ajouté à l'organigramme. A l'intérieur de ce losange, un point d'interrogation indique qu'une question est posée. Tant que la réponse est oui, le pointeur boucle par l'intermédiaire de GTO. Mais si la réponse change, le pointeur sort de la boucle et exécute une autre portion de programme. Pendant l'exécution d'un test, deux valeurs sont comparées et la question posée peut varier : la première donnée est-elle égale à la deuxième ? Est-elle différente ? Est-elle plus grande ou égale ? Plus petite ou égale ?

Dans la pratique, sur la TI 57 ces deux éléments de comparaison sont contenus l'un à l'affichage, l'autre dans une mémoire spéciale : le registre t. En fait, c'est la mémoire n°7 qui est utilisée dans ce cas-là. Et les tests possibles sont représentés dans le tableau ci-dessous.

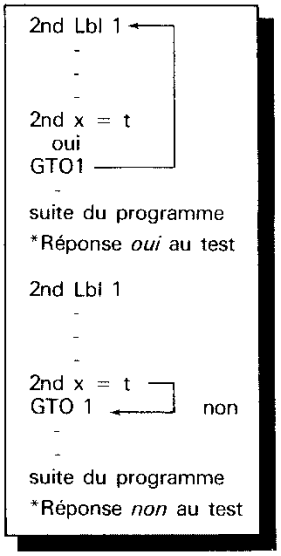
x est-il égal à t	$x = t$	noté sur la TI57	2nd $x = t$
x est-il différent de t	$x \neq t$	noté	INV 2nd $x = t$
x est-il supérieur ou égal à t	$x \geqslant t$	noté	2nd $x \geqslant t$
x est-il inférieur à t	$x < t$	noté	INV 2nd $x \geqslant t$

La réponse à la question posée peut être oui ou non. Si c'est " oui " le pas qui suit le test est exécuté. Il contient pratiquement toujours un branchement obligé qui assurera soit le bouclage du programme soit une autre opération. Si la réponse est négative, le pas sui-

La programmation : les boucles et les tests

vant le test est sauté et le pointeur suit son petit bonhomme de chemin comme si rien ne s'était passé.

Revoyons cela de façon un peu schématique :



Le registre t peut être chargé de différentes façons avant le test. L'accès normal est la boucle x ⇌ t qui échange le contenu de l'affichage et celui de t. Le résultat serait

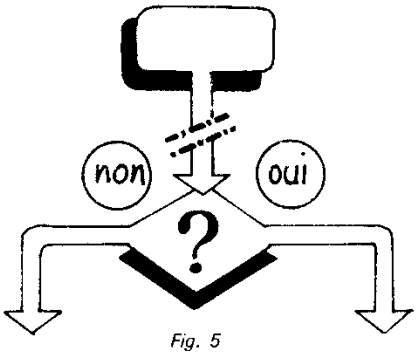


Fig. 5

le même en faisant 2nd Exc 7. Dans certains cas, il est utile de faire disparaître l'ancien contenu de t au lieu de le placer à l'affichage. Il faut alors simplement exécuter STO7.

Essayons maintenant de concrétiser tout cela avec un exemple de programme. Nous allons reprendre la progression arithmétique et rajouter un test pour obtenir un arrêt automatique d'exécution. Nous fixerons avant le départ une valeur limite qui provoquera la sortie de la boucle. Cette limite n'étant pas nécessairement un terme de la progression, nous exécuterons un test supplémentaire (voir fig. 4).

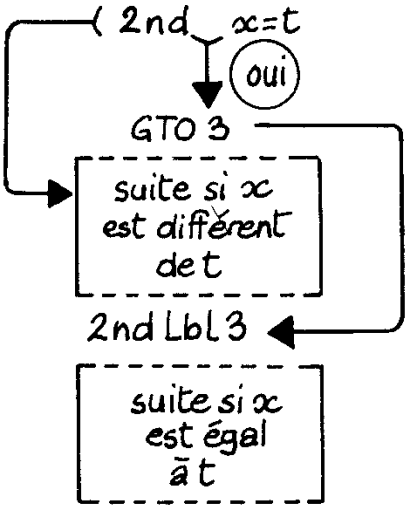


Fig. 6

Vous remarquez que le début du programme est semblable au précédent. Mais après l'affichage du terme de la progression qui vient d'être calculé, le premier test examine s'il est égal à la limite. Si c'est le cas, ce terme est affiché à nouveau, suivi par le nombre de tours de circuits parcourus. S'il n'y a pas égalité, un second test vérifie si le dernier terme obtenu est plus petit que la limite. Si la réponse est oui, un GTO réexpédie le pointeur au début de la boucle. Si la réponse est non, le dernier terme est affiché, puis la limite prévue de la progression et le nombre d'itérations effectuées. Ce deuxième test assure la sortie de la boucle en prévenant que la limite choisie ne fait pas partie de la progression.

Vous pouvez constater que le deuxième test contrôle si x est plus petit que t et non pas s'il est inférieur ou égal à t, comme cela est indiqué sur la touche : x ⩾ t. Ce résultat est obtenu par la succession des deux tests, le premier éliminant le cas de la stricte égalité, le second

00	32 0	STO 0	— Mise en mémoire de la raison
01	81	R/S	Arrêt, entrée du 1er terme de la progression
02	32 1	STO 1	Mise en mémoire
03	81	R/S	Arrêt. Chargement du registre de test avec
04	32 7	STO 7	la valeur limite de la progression
05	00	0	Remise à zéro du registre compteur
06	32 2	STO 2	de tour
07	86 1	2nd Lbl 1	
08	01	1	Comptage des itérations
09	34 2	SUM 2	
10	33 0	RCL 0	
11	34 1	SUM 1	Calcul du terme suivant de la progression
12	33 1	RCL 1	et affichage pendant
13	36	2nd Pause	une pause
14	66	2nd x = t	— Le terme est-il égal à la limite ?
15	51 2	GTO 2	— si oui aller à Lbl 2
16	-76	INV 2nd x ⩾ t	— si non est-il plus petit que la limite ?
17	51 1	GTO 1	— si oui boucler en allant en 1
18	33 1	RCL 1	— si non afficher le dernier terme calculé
19	36	2nd Pause	pendant une pause
20	22	x ⇌ t	Rappel du registre t :
21	36	2nd Pause	Afficher la limite choisie (qui dans ce cas ne fait pas partie de la progression)
22	33 2	RCL 2	nombre d'itérations effectuées
23	81	R/S	
24	86 2	2nd Lbl 2	
25	33 7	RCL 7	Afficher le dernier terme calculé
26	36	2nd Pause	
27	33 2	RCL 2	Afficher le nombre d'itérations
28	36	2nd Pause	Le 2nd Pause pourrait être supprimé, mais
29	81	R/S	il permet de différencier les deux circuits.

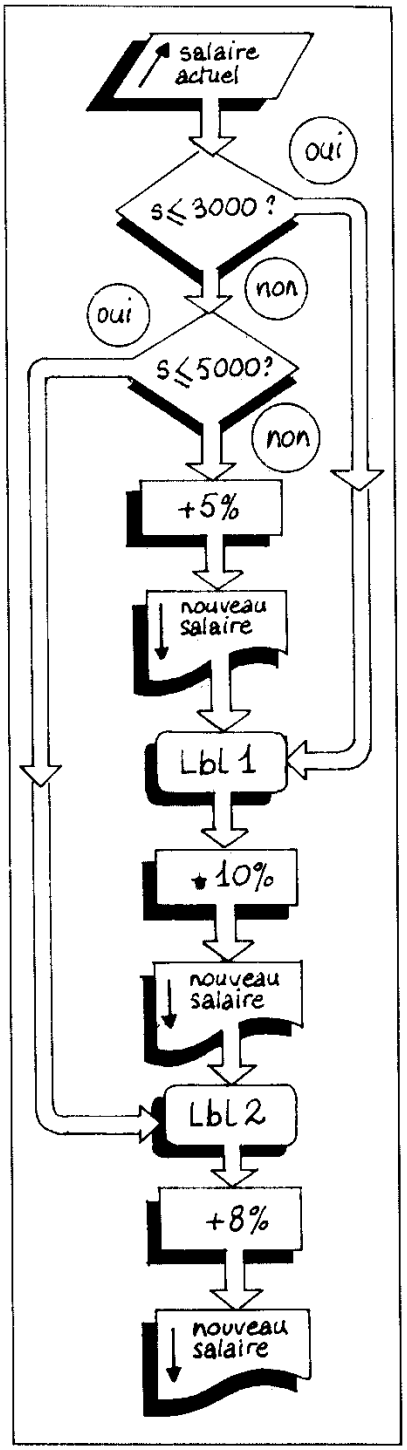


Fig. 7

00	22	x ⇌ t	Entrée du salaire actuel dans registre t
01	03	3	
02	00	0	3 000
03	00	0	
04	00	0	
05	76	2nd x ≥ t	Est-ce plus grand que le salaire ?
06	51 1	GTO 1	— oui : aller à 1 ; sinon, on poursuit.
07	05	5	
08	00	0	
09	00	0	5 000
10	00	0	
11	76	2nd x ≥ t	Est-ce plus grand que le salaire ?
12	51 2	GTO 2	— oui : aller en 2.
13	33 7	RCL 7	— non : on calcule donc le nouveau
14	75	+	salaire avec une augmentation de 5 %
15	14	CE	
16	55	x	
17	83	.	
18	00	0	
19	05	5	
20	85	=	
21	81	R/S	
22	86 1	2nd Lbl 1	Calcul du nouveau salaire dans le cas où
23	33 7	RCL 7	l'ancien était inférieur à 3 000
24	75	+	
25	14	CE	
26	55	x	
27	83	.	
28	01	1	
29	85	=	
30	81	R/S	
31	86 2	2nd Lbl 2	Calcul du nouveau salaire lorsque l'ancien
32	33 7	RCL 7	était compris entre 3 000 et 5 000 F.
33	75	+	
34	14	CE	
35	55	x	
36	83	.	
37	00	0	
38	08	8	
39	85	=	
40	81	R/S	

Charger le programme, faire RST. Entrer l'ancien salaire et R/S.

ne peut plus contrôler que la stricte inégalité.

Nous n'avons jusqu'ici considéré les tests que comme un moyen de sortir d'une boucle de programme. Vous avez sans doute constaté, dans le programme précédent, qu'ils peuvent servir à autre chose : proposer des voies différentes pour la suite du programme selon le résultat du test. Dans un cas une suite d'opérations sera effectuée, dans l'autre la succession sera différente, ce qui est schématisé à la figure 5.

Dans la pratique du programme cela se traduira très souvent par un GTO après le test envoyant le pointeur à une étiquette située plus bas dans le programme. Entre ce GTO et l'étiquette en question, l'autre voie de réponse au test est représentée par la figure 6.

Nous allons détaillé ce cas dans un exemple.

Un patron décide d'augmenter ses employés, en faisant varier le taux d'augmentation en fonction du salaire de chacun (fig. 7).

Il donne 10 % à ceux dont le salaire net est inférieur à 3 000 F, 8 % si le salaire est compris entre 3 000 et 5 000 F, et 5 % pour les rémunérations supérieures à 5 000 F. Le programme devra calculer le salaire après augmentation.

Vous rencontrerez dans ce programme une utilisation un peu particulière de la touche CE. Elle sert habituellement à effacer le registre d'affichage si il n'y a pas eu de signe opératoire appuyé. Ici, elle se comporte comme un rappel derrière le signe opératoire du contenu du registre d'affichage.

Essayez la séquence de touches 5,0,0, + , CE, = , le résultat est 1 000 ce qui montre que l'appui de CE a maintenu 500 à l'affichage pour l'opération.

Vous pouvez également constater que la même séquence de calcul est répétée 3 fois, avec relativement peu de différence (seul le taux d'augmentation varie). Cela représente un gâchis important de place en mémoire programme. Il est heureusement possible d'éviter cette réécriture en utilisant les sous-programmes. Et nous en parlerons une prochaine fois.

☐ Xavier de La Tullaye

Plus facile que le 21 : voici le 13

Voici en 37 pas un programme pour votre TI-57. C'est une variante du jeu de 21, baptisée « le 13 ». Porte-bonheur ?

■ Les règles sont simples : vous jouez en premier en tirant une à une, et bien sûr au hasard, des cartes dont la valeur est comprise entre 0 et 9. A chaque tirage, votre total augmente de la valeur de la carte, et vous pouvez tirer autant de cartes que vous le voulez. Mais si votre total dépasse 13, vous avez perdu la partie.

Quand votre total vous satisfait (par exemple 11, 12 ou même 13 si vous êtes chanceux), vous conservez votre score et le programme joue à son tour en essayant de faire mieux ou aussi bien que vous, bien entendu sans dépasser le chiffre fatidique de 13.

Vous pouvez suivre sur l'affichage de la TI-57 la progression du score de votre adversaire jusqu'au moment où il s'arrête : le résultat de la partie est alors affiché.

Les cartes sont évidemment remplacées par un générateur de nombres pseudo-aléatoires qui travaille honnêtement.

Quel est donc le mode d'emploi du programme

1. On commence par donner une " source " au générateur de nombre aléatoires en stockant dans le registre 2 une valeur comprise entre 0 et 1 (par exemple : 0.637 STO 2).

2. Le joueur débute la partie : il appuie sur RST (retour du pointeur au pas 00) puis sur R/S. Les points qu'il a obtenus apparaissent à l'affichage. S'il désire tirer d'autres cartes, il appuie encore sur R/S, et son nouveau total apparaît. Si l'affichage est clignotant, la partie est terminée : son total a dépassé 13, et le joueur a perdu.

3. S'il décide de s'arrêter sans avoir dépassé 13, c'est à la machine de jouer : on appuie sur SBR1 et elle affiche régulièrement ses totaux successifs (grâce à l'instruction pause du pas 21) puis s'arrête.

4. On peut alors lire les résultats qui sont toujours affichés clignotants :

- avec 9.9999999 99, le match est nul ;
- dernier total de la machine : l'adversaire humain a gagné, car le résultat de la machine dépasse 13 ;
- tout autre nombre indique que le programme est vainqueur.

Pour faire une autre partie, il suffit d'appuyer sur CLR et de reprendre comme en 2 par RST puis R/S.

utilisez les hasards du programme

Le générateur de nombre aléatoire est situé sous l'étiquette 9 (pas 27 et suivants). Pour pouvoir l'utiliser dans un autre programme, on remplacera le RCL 7 du pas 28 par " 29 ". Ce générateur utilise la mémoire 2 qui doit contenir au départ un nombre compris entre 0 et 1 exclus, et il retourne un nombre d'un chiffre, le registre 2 contenant alors une nouvelle valeur x telle que 0<x<1. Je me suis amusé à tester sur un ordinateur individuel la répartition des tirages et, sur plusieurs milliers de nombres successifs, je n'ai constaté aucune périodicité, contrairement à ce qui se produit sur beaucoup de générateurs de ce type.

Comme cette séquence n'utilise qu'une dizaine de pas sur la TI-57, je pense qu'elle pourra vous être utile.

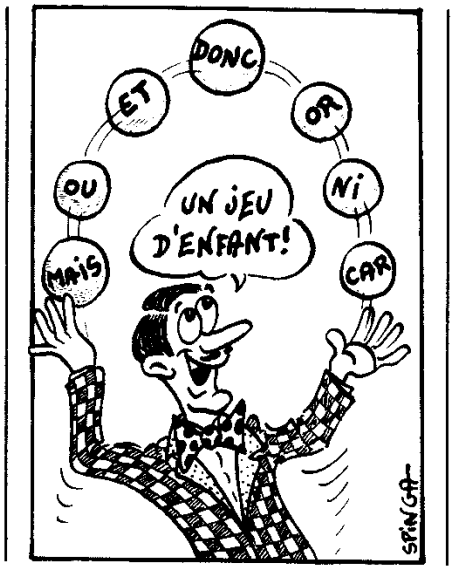
Une dernière remarque : les instructions GTO 2 des pas 12 et 23 ne renvoient à aucune étiquette LBL 2 ; c'est " étudié pour " puisqu'elles servent à provoquer l'erreur qui, la partie terminée, interrompt l'exécution du programme et provoque le clignotement de l'affichage.

☐ Christophe Théron

Liste du programme

```
      JEU DU 13
  AUTEUR : CHRISTOPHE THERON
  COPYRIGHT L'ORDINATEUR
     DE POCHE  ET L'AUTEUR
*************************
 0   15        CLR
 1   32  0     STO  0
 2   01        1
 3   32  1     STO  1
 4   01        1
 5   04        4
 6   22        X ECHANGE T
 7   86  0     2ND LBL  0
 8   61  9     SBR  9
 9   34  0     SUM  0
10   33  0     RCL  0
11   76        2ND X>=T
12   51  2     GTO  2
13   81        R/S
14   51  0     GTO  0
15   86  1     2ND LBL  1
16   61  9     SBR  9
17   34  1     SUM  1
18   33  0     RCL  0
19   65        -
20   33  1     RCL  1
21   36        2ND PAUSE
22   76        2ND X>=T
23   51  2     GTO  2
24   85        =
25   13        INX
26   51  1     GTO  1
27   86  9     2ND LBL  9
28   33  7     RCL  7
29   39  2     2ND PRD  2
30   33  2     RCL  2
31   -49       INV 2ND INT
32   32  2     STO  2
33   42        EE
34   01        1
35   -42       INV [illegible] EE
36   49        2ND INT
37   -61       INV SBR
```

M0 : score du joueur
M1 : score de la machine
M2 : nombre source pour le générateur de nombres aléatoires



Avec des si... la logique du PC 1211 TRS 80 Pocket

Si vous laissez aux spécialistes le soin de se poser les questions dont vous n'avez rien à faire, vous verrez que la logique n'a rien de sorcier. Logique, non ?

■ Parfois, lorsqu'on s'initie à l'informatique, on se fait une montagne des mathématiques qu'on pense devoir apprendre (ou réapprendre) avant d'utiliser vraiment un ordinateur. Quelques jours ou quelques semaines après, on se dit que c'était peu de choses, sans bien s'apercevoir d'ailleurs que l'on a beaucoup appris entre-temps. Il se peut même que les mathématiques, d'abord redoutées, soient devenues une occupation amusante : ce que l'on fait par plaisir ne coûte pas.

Il en va de même pour la logique. Contrairement à une idée répandue, la logique n'est pas réservée à un petit cercle de spécialistes et chacun peut s'y adonner pendant ses loisirs.

Si vous utilisez depuis quelque temps un PC 1211 ou un TRS 80 Pocket, vous connaissez déjà les fonctions logiques de votre machine, mais peut-être ne connaissez-vous pas bien les services qu'elles peuvent vous rendre lorsque vous programmez. Les notices d'utilisation (Sharp comme Tandy) n'en disent en effet qu'un strict minimum. C'est assez étonnant quand on sait qu'en réalité la machine est de ce point de vue fort bien pourvue.

Vrai ou faux

Nous nous contenterons bien entendu des rudiments de la logique. Les rudiments ont tout pour plaire : c'est sur eux que tout est construit, ils ont l'avantage d'être simples, et dans le cas présent eux seuls sont utiles pour programmer.

Quand on compare deux valeurs numériques (disons x et y), on ne peut aboutir qu'à trois conclusions dont chacune exclut les deux autres.

La première conclusion possible est que ces deux grandeurs sont égales. Si x vaut 30 et y vaut 30, nous pouvons affirmer qu'x est égal à y. Dès que cette égalité est reconnue, nous ne pouvons plus dire " x est plus grand qu'y ", car ce serait contradictoire : deux grandeurs ne peuvent pas être à la fois égales et différentes. Pour la même raison, nous ne pouvons plus dire non plus " x est plus petit qu'y " puisque nous savons qu'ils ont la même valeur.

La deuxième conclusion possible est qu'x est plus petit qu'y, ce qui entraîne nécessairement qu'y est plus grand qu'x. Cela se produit par exemple si x est égal à 15 et y à 37. Si tel est le cas, le même principe de non-contradiction nous interdit de dire qu'x est égal ou supérieur à y.

Dernière conclusion possible enfin : x est plus grand qu'y, autrement dit y est plus petit qu'x. C'est ce qui se produit par exemple lorsqu'x est égal à 13 et y à 3.

Avec des si...
la logique du PC 1211
TRS 80 Pocket

En mathématique, comme chacun sait, l'égalité se note au moyen du signe =, et l'on écrira x = y pour signifier que ces deux variables ont la même valeur. On utilise le signe < pour marquer l'infériorité : x < y se lit " x est plus petit qu'y ". C'est ainsi qu'on notera 4 < 12503. Enfin, la supériorité possède elle aussi son signe, et pour exprimer qu'x est plus grand qu'y, on écrira " x > y ".

Ces trois conclusions possibles à la comparaison de deux grandeurs sont on ne peut plus nettes. La négation de chacune d'entre elles est beaucoup moins précise. Si l'on vous dit par exemple que la valeur x n'est pas plus petite qu'y, on ne vous dit pas si elle lui est égale ou supérieure, on exclut seulement qu'elle lui soit inférieure.

En indiquant qu'x est différent d'y, on affirme que l'égalité de ces deux valeurs ne se vérifie pas, mais sans préciser laquelle des deux valeurs est plus grande que l'autre. C'est le signe <> (plus grand ou plus petit, c'est-à-dire différent) qu'on utilise dans le BASIC de Sharp pour exprimer ce type d'affirmation : x <> y se lit " x est différent d'y ".

On peut également être amené à dire qu'x est inférieur ou égal à y et l'on utilise alors la notation " x < = y ". Cela revient à dire qu'x est au plus égal à y, et la seule chose qu'on affirme en fait est qu'x n'est pas plus grand qu'y.

Troisième et dernière combinaison possible : x est plus grand ou égal à y, autrement dit " x est au moins égal à y " qui se notera x > = y. Ici, la seule possibilité exclue est, bien entendu, qu'x soit inférieur à y.

A la question de savoir quelles sont les relations de grandeur qui peuvent exister entre deux valeurs que l'on compare, il n'y a donc que six réponses possibles dont les trois premières sont tout à fait précises (infériorité, supériorité, égalité) et les trois autres incomplètes :

. x est différent d'y (on ne précise pas s'il lui est inférieur ou supérieur)
. x n'est pas plus grand qu'y
. x n'est pas plus petit qu'y.

Les notations qu'on utilise pour ces six affirmations sont récapitulées dans le tableau ci-dessous. Il n'y a rien à y ajouter, et elles suffisent pour décrire toutes les relations possibles entre deux grandeurs. Si par exemple x n'est ni plus grand ni plus petit qu'y, c'est qu'il est égal à y (cas n°2 du tableau). S'il n'est ni supérieur ni égal à y, c'est qu'il lui est inférieur (cas n°1). S'il n'est pas plus grand qu'y, il lui est inférieur ou égal (cas n°5), etc.

1. <	infériorité	2 < 75	x < y
2. =	égalité	10 = 5+5	x = y
3. >	supériorité	22 > 3	x > y
4. <>	différence (A)	8 <> 5	x <> y
5. <=	plus petit ou égal (B)	3 <= 20	x <= y
		4 <= 4	x <= y
6. >=	plus grand ou égal (C)	17 >= 15	x >= y
		39 >= 39	x >= y

(A) n'est pas égal, (B) n'est pas plus grand, (C) n'est pas plus petit.

Les six relations possibles entre deux valeurs

Ces six relations suffisent en fait pour exploiter les fonctions logiques des ordinateurs. Elles sont toutes disponibles dans le langage BASIC du PC 1211/TRS 80 Pocket.

Lorsqu'on propose à l'ordinateur l'une des six relations logiques que nous venons de passer en revue, la machine examine si l'affirmation est exacte. Si c'est bien le cas, elle lui affecte la valeur 1, c'est-à-dire qu'elle la remplace par 1. Ainsi, écrire 12 = 12 ou écrire 1 est strictement équivalent dans la logique de la machine. Si vous ne l'avez pas déjà vérifié, nous allons voir comment le faire. Mais auparavant, il n'est pas inutile de s'attarder pendant quelques lignes sur une distinction très importante à faire si l'on veut comprendre le langage BASIC.

En raison d'une ambiguïté que présente la plupart des versions du BASIC et qui d'ailleurs déconcerte presque tous les débutants, une précaution s'impose lorsque la relation utilisée est une relation d'égalité *et que son premier terme est une variable* (A, B, C ou A (27), etc.). Sur la machine dont nous parlons, cette remarque vaut lors d'une opération au clavier (mode RUN) mais aussi, la plupart du temps, pendant l'exécution d'un programme. En effet, s'il rencontre une séquence de ce type, A = 5 par exemple, l'ordinateur ne compare pas le contenu de la mémoire A et le nombre 5, il fait tout autre chose : il range 5 dans la mémoire A. C'est ce que l'on appelle une affectation. La forme explicite en est LET A = 5 (en anglais, " que la variable A soit égale à 5 "). Malheureusement, le LET est le plus souvent facultatif et il est de moins en moins employé. Pour que l'expression A = 5 ou B = C soit bien interprétée par la machine comme une question posée : " A est-il égal à 5 ? ", il est indispensable qu'elle soit inscrite entre parenthèses. On y veillera donc dans les exemples qui suivent.

Lorsque le poquette est en mode RUN et qu'on lui propose 12 = 3 * 4, ce qui est une affirmation vraie, l'enfoncement de la touche ENTER provoque l'affichage de la valeur 1. De la même façon, (25 = 5 * 5) * 7 vaut 7, car l'expression entre parenthèses étant vraie, elle vaut 1 : son produit par 7 est donc bien égal à 7. Et l'on ne s'étonnera pas que l'ordinateur affiche 0.5 lorsqu'on lui propose (30 > 15)/2 ENTER. C'est dire qu'il est parfaitement possible d'utiliser la valeur logique des expressions dans des calculs et, pourquoi pas, comme numéro de transfert après GOSUB ou GOTO : nous y reviendrons.

Lorsque l'expression qu'on lui propose est fausse (12 = 200, 3 >

25, 7 < 0, etc.), l'ordinateur lui affecte la valeur zéro. Vous vous apercevrez ainsi que 14 + (13 = 27) ENTER conduit à l'affichage de 14, car l'expression entre parenthèses étant fausse, elle ne vaut rien. Vous pouvez aussi vérifier que 10 = 15 vaut bien zéro, tout comme 666 < 21.5. En revanche, 14 + (27 > 13) et 14 + (13 < 27) valent chacun 15, (12 = 12) + (14 > 12) vaut 2, mais (14 < = 2) + (150 = 13.01) est égal à zéro, etc. On peut donc additionner les unes aux autres les valeurs logiques (0 ou 1) de plusieurs expressions. Comme nous le verrons, c'est une possibilité très intéressante quand elle est bien exploitée.

Si nous essayons maintenant d'obtenir la valeur de 23/(12 = 159), nous obtiendrons très normalement le message d'erreur du premier type qui nous signale que nous venons de demander une division par zéro. Nous pouvons également vérifier que la valeur de 45 * (59 > 2501) est nulle, ce qui n'a rien d'étonnant puisque la machine a remplacé l'expression (59 > 2501) par sa valeur logique : zéro. Si l'on y réfléchit, cette dernière remarque est très importante. Ainsi, quand on multiplie entre elles les valeurs logiques de plusieurs expressions, il suffit que l'une d'entre elles soit fausse pour que le produit de l'ensemble soit nul. Nous verrons d'ici peu le parti qu'on peut en tirer.

Avant de poursuivre toutefois, il convient de rappeler que la plupart des machines offrent des facilités de calcul vraiment pratiques mais parfois très dangereuses. L'ordinateur de chez Sharp/Tandy ne fait pas exception, et si l'on n'est pas averti de telle ou telle de ses particularités on a toutes les chances de se retrouver un jour ou l'autre devant un résultat insolite.

En réalité, on se trompe lourdement lorsqu'on s'imagine que les ordinateurs effectuent correctement toutes les opérations qu'on leur demande. Il n'en est rien. Et c'est d'autant plus insidieux que le plus souvent toutes les apparences de l'exactitude sont réunies. Ainsi, si l'on demande au PC 1211 d'élever 5 au carré (5 Λ 2 ENTER), le résultat est apparemment impeccable : 25. En réalité l'ordinateur, ou plus exactement ses constructeurs sauvent . face, car le nombre affiché (il est bien égal au carré de 5) n'est pas tout à fait celui qui a été calculé. Le nombre que la machine a calculé, c'est 24.9999999996 : il y a donc une différence qui, pour être toute petite, n'en constitue pas moins une erreur. Et le plus embêtant de l'affaire, c'est que l'ordinateur ne connaît que le résultat faux et pas celui — juste — qu'il a pourtant affiché ! Il est facile d'en apporter la preuve.

Soyons très précis

Si nous demandons, en mode RUN, 25 — (5 Λ 2) ou plus simplement 25 — 5 Λ 2, nous obtenons hélas 4.E — 10, ce qui n'est pas beaucoup, me direz-vous, puisqu'il ne s'agit que de la deux milliards cinq cent millionnième partie de l'unité (une broutille !), mais c'est amplement suffisant pour empêcher d'utiliser la fonction élévation à une puissance dans une expression logique. A la question 25 = 5 Λ 2, le PC 1211 répond 0, c'est-à-dire non. Même réponse pour 9 = 3 Λ 2, 16 = 4 Λ 2, etc. Persistant dans son erreur, l'ordinateur vous annoncera par ailleurs que la partie entière du carré de 5 : INT (5 Λ 2) est égale à 24 ! Les utilisateurs d'autres machines pourront d'ailleurs vérifier qu'ils sont, eux aussi, à la merci de surprises du même genre. Il faudra donc se garder soigneusement d'utiliser ce type de fonctions préprogrammées à l'intérieur de comparaisons logiques ; on écrira plutôt 25 = 5 * 5, 27 = 3 * 3 * 3, etc. qui fonctionnent à merveille.

Tout ce que nous venons de voir en mode RUN concernant l'estimation des expressions logiques par le poquette de Sharp et de Tandy s'applique également dans les programmes que vous pouvez réaliser : on peut très bien faire exécuter des opérations sur des valeurs logiques.
10 : A = (B > 10) * (C > 10) ou mieux :
10 : LET A = (B > 10) * (C > 10)
affectera 1 à la variable A si et seulement si B et C sont tous les deux supérieurs à 10. Dans tous les autres cas, A vaudra 0.
10 : LET A = (B > 10) + (C > 10)
affectera à la variable A une valeur comprise entre 0 et 2 selon qu'une seulement ou les deux expressions seront fausses ou vraies.
10 : GOTO (B = C) + 11
effectuera un branchement à la ligne 11 ou 12 selon que B sera ou non différent de C, ce qui constitue bel et bien un branchement conditionnel.

Exemple d'utilisation des valeurs logiques dans des branchements multiples

Lorsque le déroulement d'un programme doit dépendre de la vérité de deux propositions logiques (nous les appellerons a et b), il faut ménager 4 branchements distincts correspondant aux 4 situations possibles :
. a et b sont faux
. a seul est vrai
. b seul est vrai
. a et b sont vrais.

Si la valeur logique des propositions a été préalablement rangée dans les variables A et B par exemple, ce branchement multiple s'effectura simplement, exemple :
100 : GOTO 110 + 10 A + 20 B

Cette ligne renverra en effet :
. en 110 si a et b sont faux
. en 120 si a seul est vrai
. en 130 si b seul est vrai
. en 140 si a et b sont vrais.

Le même raisonnement appliqué à une situation où 4 propositions logiques entrent en ligne de compte conduira à programmer :

100 : GOTO 110 + 10 A + 20 B
+ 40 C + 80 D
qui effectura les 16 transferts possibles aux lignes comprises entre 110 (tout est faux) et 260 (tout est vrai).

On peut d'ailleurs faire encore plus concis en n'affectant pas la valeur logique de chacune des expressions à des variables ; dont l'ordinateur évaluera la valeur. On aura ainsi par exemple :

100 : GOTO 110 + 10* (X>=100)
+ 20* (Y=88) + 40*(Z<>2)

On y gagnera parfois de la place mémoire. On écrira alors directement les expressions et l'on se retrouvera alors en 110 (tout est faux), en 120 (X vaut au moins 100, mais Y est différent de 88 et Z est égal à 2), en 130 (X est plus petit que 100, Y est bien égal à 88 mais Z vaut 2), en 140 (X vaut au moins 100 et Y est égal à 88, mais Z n'est pas différent de 2), etc.

Avec des si...
la logique du PC 1211
TRS 80 Pocket

Ordinairement, ce type de branchement ne s'effectue pas du tout de la sorte : on utilise plutôt l'instruction prévue à cet effet, IF... THEN, c'est-à-dire " si ... alors ".

```
10 : INPUT A
20 : IF A > 10 THEN 40
(ou 20 : IF A > 10 GOTO 40)
30 : PRINT " A < = 10 " : GOTO 10
40 : PRINT " A > 10 " : GOTO 10
```

Le fonctionnement de cette instruction, qui est par excellence l'instruction logique, n'a rien de mystérieux. Lorsque l'ordinateur rencontre une instruction IF, il examine la valeur de ce qui suit immédiatement cette instruction dans la ligne du programme, que ce soit un nombre, une variable ou une expression. Au besoin, il calcule cette valeur ou il l'évalue. Si cette valeur est positive (0.000000001 par exemple, ou 129 Λ 14, ou 1, ou 7.67, etc.), il exécute la ou les instructions qui suivent sur la même ligne de programme. Dans le cas contraire, autrement dit si cette valeur est inférieure ou égale à zéro, il ignore le restant de la ligne et il saute à la ligne suivante.

```
10 : IF 0.0000001 THEN 100
```

renverra imperturbablement à la ligne 100.

```
10 : IF — 0.000003 THEN 100
20 : BEEP 2 : END
```

conduira en revanche à l'exécution de la ligne 20 où le programme s'achève après l'émission de deux signaux sonores.

Si l'instruction IF est suivie par une variable, la machine examine quel en est le contenu et elle réagit de la même façon. Si ce contenu est positif, elle exécute les dernières instructions de la ligne, sinon elle les omet pour poursuivre le programme à la ligne suivante. On peut donc écrire IF A THEN 100 ; le transfert en 100 ne s'effectuera que si le contenu de la variable A est plus grand que zéro.

Lorsqu'une expression logique suit IF, c'est la valeur de cette expression (vrai ou faux, c'est-à-dire 1 ou 0) qui détermine la façon dont se poursuit le programme. Mais rien n'empêche de combiner plusieurs expressions logiques :

```
10 : IF (A < B) + (B < C) then 200
```

effectuera le transfert à la ligne 200 si l'une ou l'autre des deux expressions est vraie. En effet, 1 + 0 = 1 et 0+1 = 1. Mais le transfert se fera aussi si les deux conditions sont vraies, car 1 + 1 = 2, ce qui est encore une grandeur positive. Si les deux sont fausses, le branchement ne s'effectue pas : 0 + 0 = 0. C'est aussi simple que cela.

Ainsi, lorsqu'on additionne des expressions logiques derrière l'instruction IF, il suffit que l'une d'elle au moins soit vraie pour que la condition soit remplie. Le signe de l'addition correspond alors au ou et plus exactement au ou dit inclusif : si plusieurs expressions sont vraies simultanément, la condition est également satisfaite. Et rien n'empêche d'additionner plus de deux expressions logiques.

Pour obtenir que la condition introduite par IF ne soit satisfaite que si l'une seulement des expressions logiques est vraie, on écrira :

```
IF ((A = B) + (A > C) = 1)
THEN...
```

et l'on parlera dans ce cas de ou exclusif. Il est naturellement possible d'obtenir aussi que le déroulement du programme soit subordonné à la vérité de deux expressions :

```
10 : IF ((A > B) + (B > C) = 2)
THEN...
```

ou de deux expressions seulement sur trois :

```
10 : IF ((A = B) + (B = C) + (B
= F) = 2) THEN...
```

Si l'on désire que la condition ne soit satisfaite que si toutes les expressions logiques sont vraies, on pourra écrire par exemple : IF ((A = C) + (B = G) + (D > F) + (H < G) = 4) THEN... mais on obtiendra le même résultat en écrivant plus simplement : IF (A = C) * (B = G) * (D > F) * (H < G) THEN... En effet, il suffira qu'un seul des termes du produit soit faux (sa valeur sera donc 0) pour que le produit lui-même soit nul. Et l'on aura ainsi, peut-être sans le savoir d'ailleurs, mis en œuvre le *et* logique.

Vous n'aurez aucune peine à trouver ce qu'il convient de programmer dans les autres cas de figures. Comme on a pu le voir, le poquette de chez Sharp n'a pas à rougir de ses possibilités logiques, loin de là. Avec un rien d'astuce, on en obtiendra ce que l'on veut.

☐ Jean Baptiste Comiti

abonnez-vous à l'Ordinateur de poche

Consultez la base

Le système décimal paraît aller de soi, mais il y en a d'autres. Comment changer de système ?

■ On nous parle souvent de bases binaire ou octale, voire hexadécimale ou plus classiquement décimale, mais il est rare que l'on nous donne *explicitement* les moyens de passer de l'une à l'autre. Par exemple, soit le nombre 901 exprimé en base 10 ; quelle est l'expression de ce même nombre en base 7 ? (j'ai toujours été beaucoup attiré par ce nombre magique qu'est le nombre 7). Plutôt que de commencer par énoncer une règle, je vais vous montrer comment je fais moi-même.

Je dispose donc d'un nombre N0 valant 901. Divisons ce nombre par 7. On obtient $901 = 128 \times 7 + 5$, où 5 est le reste de la division de 901 par 7, et 128 le quotient. Pour avoir l'air plus sérieux, je vais les appeler respectivement C0 pour le reste, et N1 pour le quotient.

Recommençons la même opération, mais en l'appliquant cette fois-ci au quotient que nous venons d'obtenir : 128. Vous devriez obtenir comme moi $128 = 18 \times 7 + 2$, d'où, vous vous en doutiez, une valeur de 18 pour N2 et de 2 pour C1. Recommençons encore, nous obtenons C2 = 4 et N3 = 2. Recommençons encore, et cette fois N4 = 0 et C3 = 2 (oui, c'est bien la même valeur que N3).

Si maintenant je réunis tous ces renseignements, j'obtiens que N0 vaut :
$(C0 + 7 \times (C1 + 7 \times (C2 + 7 \times (C3 + 0))))$
que je réécris astucieusement en sens inverse
$(((0 + C3) \times 7 + C2) \times 7 + C1) \times 7 + C0$.

En regroupant les puissances de 7, on obtient sans la moindre surprise $C3 \times 7^3 + C2 \times 7^2 + C1 \times 7^1 + C0 \times 7^0$, puisque chacun sait que 7^0 vaut 1. Comme tous les nombres C3, C2, C1 et C0 sont compris entre 0 et 6 (puisque inférieurs au diviseur 7), ils représentent bien le nombre 901 tel qu'on l'écrit en base 7, ce que l'on note habituellement $901_{10} = 2425_7$.

D'où l'algorithme simple pour exprimer dans une base donnée B un nombre entier rangé dans une mémoire N :
Algorithme A :
A1-Diviser N par B, obtenant un quotient entier Q et un reste R ; afficher R, et recopier Q dans N.
A2-Si N n'est pas nul, recommencer l'opération A1.
A3-Fin

Cet algorithme est on ne peut plus clair, il présente toutefois le " simple " petit inconvénient de donner les chiffres de la représentation en base B dans le mauvais sens, c'est-à-dire en partant du chiffre des unités suivi de celui des B-aines, au lieu de procéder en sens inverse comme il est habituel.

Les modifications à apporter à l'algorithme A ne sont pas très complexes si l'on dispose comme c'est le cas en BASIC d'un tableau indicé : il suffira de le remplir progressivement, avant de l'imprimer en sens inverse. D'où par exemple l'algorithme B.
Algorithme B :
B0-(initialisations) Mettre 1 dans le compteur J, réserver un tableau C d'une dimension suffisante.
B1-(calcul)Diviser N par B, obtenant un quotient entier Q et un reste entier R ; recopier Q dans N et R dans C (J), et ajouter 1 à J.
B2-(test) Si N n'est pas nul, recommencer l'opération B1.
B3-(impression) Imprimer les J cases du tableau C, dans l'ordre C(J), C(J-1), ..., C(1).
B4-Fin.

Dans le cas d'une calculatrice programmable, on ne dispose pas toujours d'une possibilité analogue. Mais, si la base B n'est pas supérieure à 10, il existe un moyen simple de s'en sortir pour une valeur N pas trop grande, celui présenté dans l'algorithme C.
Algorithme C :
C0-(initialisations) Mettre 1 dans la mémoire J et 0 dans la mémoire K.
C1-Diviser N par B, obtenant un quotient entier Q et un reste entier R ; recopier Q dans N, multiplier R par J et l'ajouter au contenu de K ; multiplier J par 10.
C2-Si N n'est pas nul, recommencer l'opération C1.
C3-Imprimer K.
C4-Fin.

Si la base B est supérieure à 10, on peut encore adapter l'algorithme en multipliant à l'étape C1 J par 100 au lieu de 10, mais il faudra bien faire attention à lire par tranches de deux les chiffres obtenus. Par ailleurs, dans le cas de l'algorithme C non modifié, attention si le nombre N est trop grand et/ou la base B trop petite : K risque de dépasser le nombre de chiffres significatifs de la machine, et le résultat obtenu sera d'une valeur contestable. Par exemple, si votre machine comporte 10 chiffres significatifs, vous ne pourrez pas en base 2 exprimer des nombres supérieurs à 2^{10}, soit 1024, parce que sinon ce résultat comporterait plus de 10 chiffres.

Nous avons préféré exprimer les nombres au moyen des dix chiffres arabes (0, 1, 2, 3... 8 et 9). C'est une solution qui a le mérite de la simplicité à condition que l'on fasse bien attention à ne pas interpréter les nombres ainsi écrits comme des nombres nécessairement décimaux.

Il faut d'abord consulter la base ! Nous sommes habitués à calculer en base 10 et c'est un réflexe dont on doit se méfier. Quand la base est différente de 10, le 7 de 79 ne représente pas 7 dizaines comme nous avons tendance à le penser spontanément, mais 7 fois la base à la puissance 1.

Voilà, vous savez donc convertir une valeur numérique dans n'importe quelle base. Qu'en est-il du problème inverse ? Par exemple, quelle est en base 8 l'expression de la valeur qui s'exprime par 2425 en base 7 ?

Une façon de procéder consiste à convertir cette valeur en décimal, puis à la convertir dans l'autre base. Nous savons faire cette deuxième conversion. Il ne nous reste donc qu'à effectuer l'autre.

Reprenons notre exemple : comment s'exprime en base 10 le nombre qui s'écrit 2425 en base 7 ? (Notez bien que malgré ses expressions différentes, il s'agit bien du même nombre et des mêmes valeurs).

Pour obtenir le résultat, calculons successivement :

— $0+2=2$
— $2\times7+4=18$
— $18\times7+2=128$
— $128\times7+5=901$

et voilà notre résultat. Un algorithme pour ce calcul pourrait être :
Algorithme D : à partir des chiffres représentant un nombre N en base B, calcule la valeur de ce nombre et le donne en représentation décimale.
D0 - (initialisations) Mettre 0 dans M.
D1 - Multiplier M par B. Lire le chiffre suivant de N (en partant de la gauche), et l'ajouter à M.
D2 - (test) Si tous les chiffres de N n'ont pas été traités, revenir à l'étape D1.
D3 - Afficher M.
D4 - Fin.

Suivant la machine dont vous disposez, la lecture des chiffres de N aux étapes D1 et D2 sera plus ou moins simple. Mais cet algorithme fonctionne, et qui plus est avec le nombre minimum de multiplications et d'additions (1).

Nous savons donc maintenant convertir un nombre entier d'une base à l'autre sans aucun problème. Qu'en est-il d'un nombre comme 12,53 (ou, comme écrivent les anglo-saxons, 12.53) ? Nous savons en fait déjà faire la moitié du travail : la conversion de la partie entière. Il ne nous reste donc plus que la conversion de la partie fractionnaire.

Nous risquons bien sûr ce faisant de rencontrer un problème : celui du nombre de « décimales », car certaines valeurs ne « tombent » pas juste. Ainsi 0.1_3 (que l'on note plus souvent 1/3 !) s'exprime-t-il en base 10 par 0.3 suivi d'un nombre infini de 3. Calculons en base 7 la valeur 0.53, en décidant de nous arrêter aux 6 premières décimales :

— $0.53\times7=3.71$, dont la partie entière nous donne 3.
— $0.71\times7=4.97$, dont la partie entière nous donne 4.
— $0{,}97\times7=6.79$, qui donne 6.
— $0.79\times7=5.53$, qui donne 5.
— $0.53\times7=3.71$, qui donne 3.
— $0.71\times7=4.97$, qui donne 4.

Nous voyons ainsi que 0.53 (base 10) s'écrit en base 7 « 0.346534_7 », suivi en fait d'un nombre infini de groupes 6534 puisque notre calcul est retombé sur les mêmes valeurs.

	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	11	13	15
3	3	6	12	15	21	24
4	4	11	15	22	26	33
5	5	13	21	26	34	42
6	6	15	24	33	42	51

Table de multiplication en base 7

	1	2	3	4	5	6
1	2	3	4	5	6	10
2	3	4	5	6	10	11
3	4	5	6	10	11	12
4	5	6	10	11	12	13
5	6	10	11	12	13	14
6	10	11	12	13	14	15

Table d'addition en base 7

Si l'on compare ce calcul à celui de la conversion des nombres entiers, on voit que « tout se fait à l'envers » ou presque : on obtient dans le calcul les chiffres dans le bon ordre, et non en sens inverse, on multiplie au lieu de diviser, on prend la partie entière au lieu du reste, et la partie décimale au lieu du quotient.

Notez également que le nombre de décimales que l'on multiplie est constant (toujours 2 chiffres après le point dans notre exemple), sauf éventuellement si 5 intervient. Ceci entraîne d'ailleurs que le développement, c'est-à-dire le nombre de « décimales » dans la nouvelle base, est en général illimité.

Donnons maintenant l'algorithme de conversion en base B d'une valeur N strictement inférieure à 1.
Algorithme E :
E1 - Multiplier N par B, obtenant une partie entière E et une partie décimale D ; afficher E, et recopier D dans N.
E2 - (test) Si N est différent de zéro, et si le nombre voulu de décimales n'a pas été affiché, retourner en E1.
E3 - Fin.

Cet algorithme très simple présente un tout petit défaut : la valeur donnée dans le cas où le calcul ne tombe pas juste est une valeur par défaut. On pourrait envisager de la donner en fait arrondie : ceci est laissé au lecteur à titre d'exercice...

En ce qui concerne le problème de la traduction par exemple en base 7 d'une partie « décimale » exprimée en base 9, le problème est en fait plus complexe que le problème similaire sur les parties entières : les erreurs dues aux nombres illimités de « décimales » peuvent se cumuler si l'on passe par l'étape intermédiaire de la conversion en base 10. La solution, c'est de considérer que dans l'algorithme E les multiplications de l'étape E1 se font *dans la base de départ.*

Ainsi, pour convertir 0.3465_7 en base 10, on note que 10 s'écrit en base 7 13_7, on fait soigneusement sur un papier annexe la table de multiplication en base 7 *(voir les tableaux ci-dessus)*, et l'on applique l'algorithme E.
Voyons ce que cela donne sur notre exemple :

— $0.3465\times13=5.2031$, qui donne 5_7, soit 5_{10}.
— $0.2031\times13=2.6433$, qui donne 2_{10}.
— $0.6433\times13=12.2262$, qui donne 12_7, soit 9_{10}.
— $0.2262\times13=3.3056$, qui donne 3.

(Rappel : tous les calculs sont faits en base 7 !).

On obtient donc 0.5293 comme expression en base 10 des 4 premières décimales de la valeur qui s'écrit 0.34650 en base 7. Donnons maintenant la conversion en base 8 de cette même valeur 0.34650_7. Bien entendu, 8_{10} s'écrit 11_7.

— $0.3465\times11=4.1445$, qui donne 4_8.
— $0.1445\times11=1.6225$, qui donne 1.
— $0.6225\times11=10.1505$, qui donne 10_7, soit 7_8.
— $0.1505\times11=1.6555$, qui donne 1.
— $0.6555\times11=10.5435$, qui donne 7_8.

Les 5 premières « décimales » en base 8 de 0.34650_7 s'écrivent donc 0.41717_8.

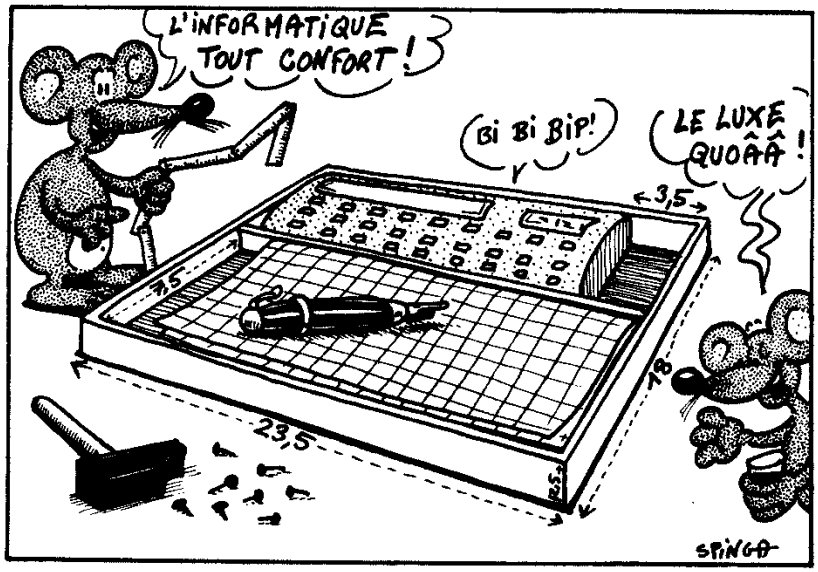
A titre d'exercices, je suggère d'étudier la représentation en base 2 de 0.01_{10}, ainsi que la modification de tous les algorithmes, donnés ici en fait pour des nombres positifs, à des nombres positifs ou négatifs.

☐ Dominique Fibo

(1) Pour plus de détails, se reporter à « Prudence avec les puissances », L'Ordinateur Individuel, *n° 25, mars 81, pp 104-105.*

Faites feu de tout bois pour l'imprimante de votre ordinateur de poche

En informatique, le bois est une matière première très rarement utilisée. Et pourtant, en voici pour votre xxxpoche.



■ Certains ordinateurs de poche peuvent être connectés à des imprimantes thermiques (HP 41C, TI 58 et 59) ou à aiguilles (PC 1211 et TRS 80 pocket) que l'on trouve dans le commerce. Il est hors de question de les fabriquer soi-même, mais elles font le travail qu'on attend d'elles toutes seules et bien.

Utilisateur d'un PC 1211, j'ai en l'idée de confectionner moi-même mon « imprimante » (vous avez remarqué les guillemets...), et même si cette dernière ne fonctionne pas de façon tout à fait autonome, elle me satisfait pleinement.

En fait, il s'agit d'un ensemble rigide comportant à la fois l'ordinateur et un bloc-notes sur lequel chacun peut écrire des textes (majuscules et minuscules descendantes, jeux de caractères personnalisés !) et même tracer des courbes en plusieurs couleurs. J'utilise personnellement un bloc Rhodia n° 8 pour ses dimensions voisines du PC 1211.

Le tout a été construit au moyen de pièces de contreplaqué (12,5 mm d'épaisseur) assemblées à l'aide de quelques clous. Si l'on est adroit de ses mains, on peut bien entendu utiliser des épaisseurs moindres, ce qui allège l'ouvrage, et, après un ponçage soigneux, passer une couche ou deux de vernis. Les ébénistes amateurs ou non mettront leur point d'honneur à ne travailler que du bois massif, à choisir une essence qui se marie bien avec les couleurs de leur machine et à effectuer la plupart des assemblages à queue d'aronde.

La profondeur des logements où viennent s'insérer le PC 1211 et le bloc-notes est de 15 mm. Si l'on désire une finition très soignée, on les capitonnera intérieurement en collant un feutre mince ou un velours.

Cette petite construction simple vous permettra de jouer au jeu de la vie, à l'avion-espion, etc... confortablement installé dans votre fauteuil préféré.

☐ Didier Bicking

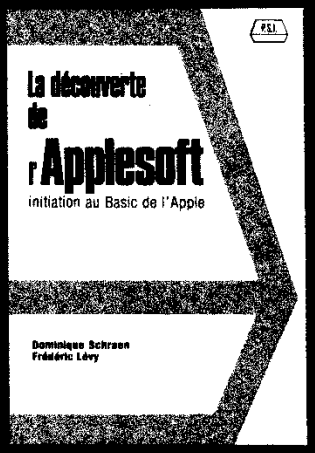
Désormais disponible*
chez votre marchand de journaux

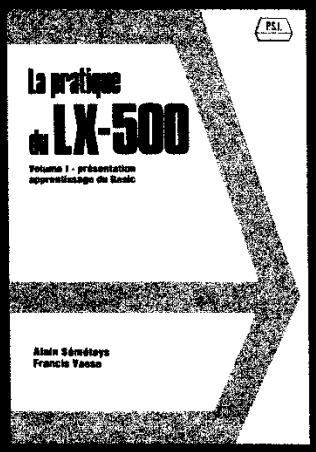
minis et micros
informatique électronique

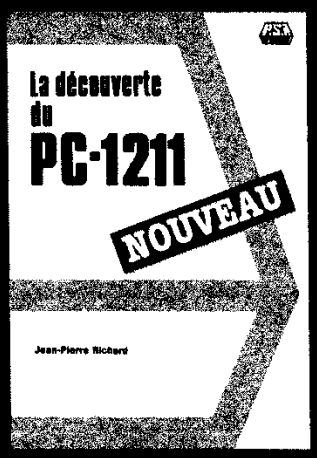
la revue professionnelle sur la technologie et les applications de la micro-informatique

*un lundi sur deux

MATERIELS







Editions du P.S.I.
41-51, rue Jacquard
BP 86 - 77400 Lagny-s/Marne
Téléphone (6) 007.59.31

au Canada : SCE Inc.
3449 rue Saint-Denis
Montréal Quebec H2X3L1
Tel. (514) 843 76.63

OP3

BON DE COMMANDE

Envoyer ce bon
accompagné
de votre règlement à
EDITIONS DU P.S.I.
41-51, rue Jacquard
BP 86 77400 Lagny-s/Marne
Tél. (6) 007.59.31.

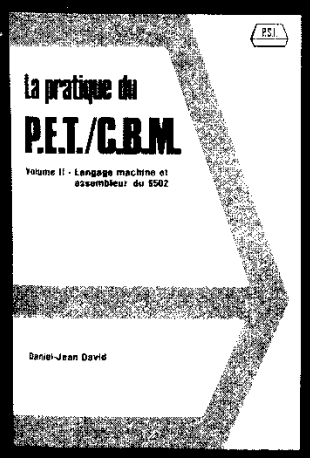
DESIGNATION	NOMBRE	PRIX
	TOTAL	

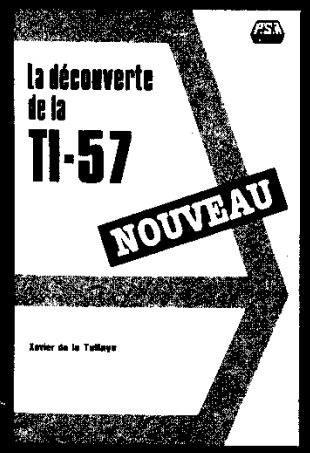
Les prix sont : taxes, emballage et port compris.
(par avion : ajouter 5 FF par livre)

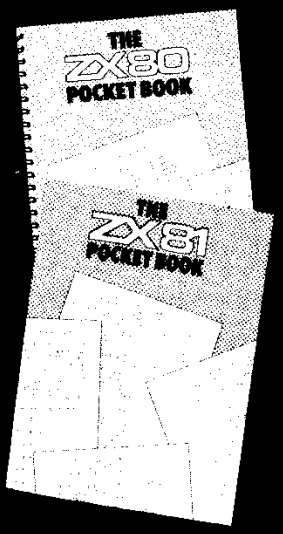
NOM _____ PRENOM _____

rue _____ N° _____

Code post. |_|_|_|_|_| Ville _____







Votre bibliothèque informatique

Les ouvrages des éditions du P.S.I. sont répartis en quatre séries de difficulté croissante : - **Série verte** : initiation - **Série bleue** : perfectionnement - **Série rouge** : approfondissement - **Série noire** : maîtrise de la technique.

La découverte de l'Applesoft
par Dominique Schraen et Frédéric Levy

Cet ouvrage d'initiation s'adresse aussi bien aux futurs utilisateurs de l'Apple voulant apprendre la programmation en Basic Applesoft, qu'à l'Applelophile chevronné sollicité par ses proches curieux de "voir un peu comment ça marche". D'approche progressive, il est illustré de nombreux exemples et exercices.

Série verte - 128 pages - 65 FF

La pratique de l'Apple II - *Volume 1*
par Nicole Bréaud-Pouliquen

Cet ouvrage présente les spécificités du Basic Applesoft à partir d'une description du matériel et du logiciel du système Apple. Les techniques de programmation, de composition et d'animation de dessins et graphiques colorés y sont expliquées à l'aide d'exemples illustratifs et d'exercices résolus.

Série bleue - 128 pages - 65 FF

La pratique de l'Apple II - *Volume 2*
par Nicole Bréaud-Pouliquen

Ce second volume de la pratique de l'Apple II est consacré au système d'exploitation disque, à la gestion des fichiers, à l'impression et aux imprimantes, à la carte horloge Appleclock. De nombreux exemples de programmes illustrent les fonctions et les commandes décrites.

Série rouge - 120 pages - 65 FF

La pratique de l'Apple II - *Volume 3*
par Nicole Bréaud-Pouliquen et Daniel-Jean David

Ce volume est une initiation à la programmation en langage machine 6502, dont le jeu d'instruction est expliqué et utilisé. L'assembleur symbolique et ses logiciels connexes y sont décrits. L'interaction avec le Basic et avec le système y sont étudiés.

Série noire - 176 pages - 75 FF

La pratique du LX 500
par Alain Séméteys et Francis Vasse

Cet ouvrage est une initiation au langage Basic et à sa programmation directement associée au LX 500 et à son logiciel. Sa démarche progressive est ponctuée d'exercices et d'exemples qui sont constitués de programmes s'enrichissant au fur et à mesure de la découverte du langage.

Série verte - 160 pages - 75 FF

La pratique du MZ-80K
par Jean-Pierre Lhoir

Cet ouvrage présente les caractéristiques du Basic SP 5025 de l'ordinateur Sharp MZ-80K. Il comprend plusieurs annexes : messages d'erreurs, code ASC II, réglages de l'écran et du haut-parleur ainsi qu'un répertoire des instructions Basic.

Série bleue - 128 pages - 65 FF

La découverte du PET/CBM
par Daniel-Jean David

Cet ouvrage d'initiation à l'utilisation de la gamme PET/CBM comprend, essentiellement, une introduction progressive au langage Basic. La découverte du langage est conduite en bâtissant des programmes de plus en plus élaborés, au fur et à mesure de l'introduction de notions nouvelles.

Série verte - 136 pages - 65 FF

La pratique du PET/CBM
Volume 1
par Daniel-Jean David

Cet ouvrage, qui suppose une bonne connaissance du Basic et des commandes du PET/CBM, ouvre les portes des applications faisant appel aux fichiers (cassettes, disquettes), à l'impression et au bus IEEE. Il comporte également de nombreux exemples et exercices avec solution.

Série bleue - 136 pages - 65 FF

La pratique du PET/CBM
Volume 2
par Daniel-Jean David

Ce volume est une initiation à la programmation en langage machine 6502, dont le jeu d'instruction est expliqué et utilisé. L'assembleur symbolique, l'éditeur et le chargeur, y sont décrits. L'interaction avec le Basic et avec le système y sont étudiés.

Série rouge - 176 pages - 75 FF

La découverte du PC-1211
par Jean-Pierre Richard

Au fil des chapitres, cet ouvrage fera découvrir au non initié instructions et commandes, variables et mémoires, fonctions périphériques. Enrichi d'exercices d'applications et d'un index, ce manuel fournit à l'utilisateur tous les éléments de base nécessaires à la programmation en langage Basic du PC-1211 (ou TRS-pocket).

Série verte - 152 pages - 75 FF

La pratique du TRS-80 - *Volume 1*
par Pierre Giraud et Alain Pinaud

Cet ouvrage s'adresse aux amateurs de TRS-80 Modèles I et III déjà initiés. Il permettra aux utilisateurs de TRS-80 de tirer le meilleur parti de leur ordinateur et, en particulier, du langage BASIC niveau II. Il est complété par des exemples de programmes et des annexes pratiques.

Série bleue - 128 pages - 65 FF

La pratique du TRS-80 - *Volume 2*
par Pierre Giraud et Alain Pinaud

Ce volume s'adresse à ceux qui s'intéressent à la programmation en langage assembleur du Z-80. L'auteur aborde l'aspect mise au point de programmes écrits en langage assembleur et cite quelques outils. Il présente enfin des exemples de sous-programmes et décrit les entrées-sorties du TRS-80.

Série rouge - 220 pages - 85 FF

La pratique du TRS-80 - *Volume 3*
par Pierre Giraud et Alain Pinaud

Après le logiciel, le matériel, ce volume vous guide dans l'exploration systématique (55 schémas) du TRS-80 Modèle I. Pour mordus sans complexe en électronique dont une des armes familières est le fer à souder. Ce livre est, par ailleurs, une documentation originale et complète sur la structure d'un ordinateur.

Série noire - 128 pages - 75 FF

La découverte de la TI - 57
par Xavier de la Tullaye

S'adressant aux débutants, cet ouvrage les conduira, dans un langage clair, de l'élémentaire 2 + 2 à des programmes perfectionnés. Après une étude fonctionnelle de la calculette, la programmation est expliquée progressivement, de la conception à la réalisation en s'appuyant sur de nombreux exemples.

Série verte - 144 pages - 65 FF

EDITIONS PHIPPS

The ZX-80 Pocket book
(in english)

128 pages - 65 FF

The ZX-81 Pocket book
(in english)

Le Sinclair ZX-81 est désormais en vente en France. Les Editions du PSI commercialisent ce livre anglais qui constitue, au-delà de la documentation du fournisseur, une mine de conseils et de trucs pour mieux utiliser votre ZX-81.

136 pages - 65 FF

Essayez de tirer un maximum

Que ce soit pour des raisons scolaires ou professionnelles, on a couramment besoin de trouver le maximum ou le zéro d'une fonction. Pourquoi ne pas le demander à un micropoche ?

■ Chaque année, 80% des candidats au baccalauréat de la section D, lorsqu'ils lisent l'énoncé des problèmes de mathématiques qui leur sont soumis, découvrent qu'ils vont devoir « plancher » sur l'étude d'une fonction : $f(x) = \log x - 1/(x-1)$ ou quelque chose d'approchant. Dans l'étude de cette fonction, il y a si je puis dire un grand moment, c'est évidemment la recherche des *maximums* et des *minimums,* c'est-à-dire la mise en évidence d'intervalles consécutifs dans chacun desquels la fonction est monotone (croissante ou décroissante).

Dans cette recherche, l'arme n° 1 est la notion de dérivée. Mais l'apparition des ordinateurs de poche a donné (par exemple au postulant bachelier) d'autres atouts. En calculant systématiquement différentes valeurs prises par la fonction :

$$\left\{f(x),\ f(x_2),\ \ldots,\ f(x_n),\ \ldots\right\}$$

il est facile, sur du papier millimétré, d'obtenir l'allure générale de la courbe représentative de la fonction et par conséquent une approximation des valeurs de x pour lesquelles f passe par un maximum ou un minimum. Cette méthode toutefois n'est pas très précise : la simplicité se paie par une qualité médiocre des résultats.

Elle permet toutefois d'isoler assez grossièrement des intervalles [a, b] dans lesquels f possède un maximum unique. Nous ne nous préoccuperons pas des minimums. En effet, si f admet un minimum dans [a, b], on appliquera la même méthode à la fonction opposée $(-f)$ qui possède alors un maximum dans cet intervalle.

Prenons un exemple. Si vous avez à étudier.

$$f(x) = \frac{x^2 - 6x + 7}{1 - x}$$

pour $1 \leqslant x \leqslant 3$, votre machine vous donnera le tableau ci-dessous.

Il est donc clair que f possède un maximum entre 2 et 3 (probablement même entre 2,3 et 2,6) voisin de 1,17. La courbe représentative est, à peu près celle de la figure 1.

Ce que nous voudrions connaître, c'est une bonne valeur de T, l'abscisse du maximum M, et — bien entendu — ce maximum lui-même.

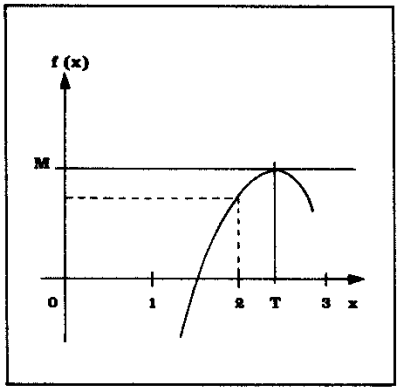


Fig 1. Le tracé point par point de la courbe fait apparaître un maximum M pour une valeur de x comprise entre 2 et 3.

De ce renseignement dépend la note du malheureux candidat bachelier !

Une étude méthodique des valeurs de f(x), aux alentours de 2,4 ou 2,5, donnerait, avec un peu de patience, que T est approximativement égal à 2,414, avec M voisin de 1,1716. Mais une telle recherche est assez ennuyeuse. Pourquoi ne pas confier, comme d'habitude, une telle tâche répétivive à notre ami l'ordinateur ? (Et encore, le bachelier n'a à étudier que deux ou trois maxi/minimums, alors que les forçats qui rédigent, en juin et juillet, les corrigés des problèmes de bac pour les éditeurs scolaires en ont à calculer une bonne centaine...)

Un de mes amis m'a signalé une méthode standard extraordinaire dont je ne connais pas l'origine (avis aux érudits...). Au lieu de procéder par tâtonnements, on se livre à une exploration systématique, un peu surprenante a priori, qui fournit toute la précision voulue. Evidemment elle ne dira pas à notre élève que T est rigoureusement égal à $1+\sqrt{2}$ et que M vaut $4-2\sqrt{2}$, ce que l'étude de la dérivée

$$f(x) = \frac{1 + 2x - x^2}{(1 - x)^2}$$

lui donnerait facilement, mais elle permettra, par exemple, de savoir que

$$2{,}4135 < T < 2{,}4146$$

ce qui permet de contrôler le travail de la dérivée !

L'étape fondamentale consiste à « resserrer » l'intervalle [a, b] en un intervalle [a', b'], contenant T et tel que :

$$a \leqslant a' < b' \leqslant b,\ b' - a' = (b - a)/2$$

Au bout de n étapes, on obtiendra

x	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	3,0
f(x)	*error*	− 16,10	− 6,20	− 2,97	− 1,40	− 0,50	0,07	0,44	0,70	0,88	1,00	1,08	1,13	1,16	1,17	1,17	1,15	1,12	1,09	1,05	1,00

un intervalle [u,v] contenant T et de largeur $v-u = (b-a)/2^n$. On prend, comme valeur approchée de T le nombre $T_0 = (u+v)/2$, comme valeur approchée de M le nombre $M_0 = f(T_0)$. On a l'inégalité

$$\left|T-T_0\right| \leqslant \frac{b-a}{2^{n+1}}$$

On rencontre souvent ce procédé (appelé *dichotomie,* littéralement « coupure en deux » en mathématiques ; mais il est ici un peu plus complexe que d'habitude. Les dichotomies usuelles donnent

$$(a'=a,\ b'=\frac{a+b}{2})$$

$$\text{ou } (a'=\frac{a+b}{2},\ b'=b)$$

Ici il y aura une troisième possibilité :

$$\left(a' = \frac{3a+b}{4},\ b' = \frac{a+3b}{4}\right)$$

Nous noterons (I) (II) (III), dans cet ordre, les trois choix possibles.

La théorie est extrêmement simple. Il n'y a en effet que quatre cas à considérer ; pour simplifier les notations, nous poserons

$$A=a,\ B=\frac{3a+b}{4},\ C=\frac{a+b}{2},$$

$$D=\frac{a+3b}{4},\ E=b$$

A C E

B D

1° cas $A \leqslant T \leqslant B$: alors f décroît sur [B,E] : $f(B)>f(C)>f(D)>f(E)$;

2° cas $B \leqslant T \leqslant C$: alors f croît sur [A,B] et décroît sur [C,E] : $f(A)<f(B)$ et $f(C)>f(D)>f(E)$;

3° cas $C \leqslant T \leqslant D$: alors f croît sur [A,C] et décroît sur [D,E] : $(f(A)<f(B)<f(C)$ et $f(D)>f(E)$;

4° cas $D \leqslant T \leqslant E$: alors f croît sur [A,D] : $f(A)<f(B)<f(C)<f(D)$.

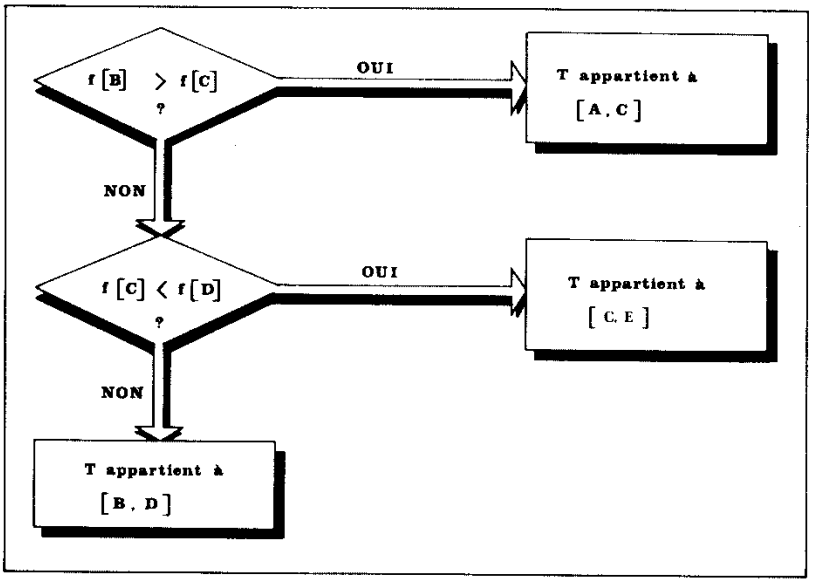
Vous trouverez figure 2, l'organigramme de l'étape centrale : on remarquera que f(A) et f(E) n'interviennent pas. Naturellement, il faut une procédure pour passer de [A,B] à un intervalle de longueur deux fois plus petite, une procédure de comptage des n opérations élémentaires ci-dessus.

Avant d'examiner les programmes, signalons que si l'on s'approche trop des limites de la machine (par exemple pour n=20 pour une HP 67 ou un PC 1211), les valeurs f(B), f(C), f(D) deviennent très difficiles à discerner les unes des autres, d'autant plus qu'on est près d'un maximum ! Il en résulte une imprécision assez grande sur T (nettement supérieure à l'incertitude théorique

$$\left|T-T_0\right| < \frac{b-a}{2^{n+1}}$$

mais (toujours parce que l'on est près d'un maximum) la précision sur M est excellente : le maximum est mieux connu que son abscisse.

Fig 2. Comment déterminer dans quel intervalle se trouve la valeur cherchée.



La liste du programme sur un " poquette " 1211.

```
 10:"A"INPUT "A=
    ";A,"B=";E,"
    N=";N
 20:FOR W=1TO N
 30:C=(A+E)/2:D=
    (C+E)/2:B=(A
    +C)/2
 40:X=B:GOSUB 15
    0
 50:Y=X:X=C:
    GOSUB 150
 60:IF Y>XGOTO 1
    00
 70:Y=X:X=D:
    GOSUB 150
 80:IF Y<XGOTO 1
    10
 90:A=B:E=D:GOTO
    120
100:E=C:GOTO 120
110:A=C
120:NEXT W
130:C=(A+E)/2:X=
    C:GOSUB 150
140:BEEP 3:PRINT
    "T= ";C:
    PRINT "MAX F
    = ";X:END
150:REM *"PLACER
     ICI LE CALC
    UL DE X=F(X)
    "
160:RETURN
```

Mode opératoire : ayant la certitude que f est d'abord croissante puis décroissante entre A et B, mettre le calcul de f en ligne 150 ; c'est ainsi que l'on aurait ici :

150 : X = (XΛ2−6*X+7)/(1−X)

Passer en mode DEF, faire SHFT A ; on lit alors "A=" et l'on tape 2 (dans l'exemple choisi, [a,b]=[2,3] puis ENTER ; on lit "B=", on tape 3 puis ENTER ; on lit "N=", on tape, par exemple, 10 puis ENTER. Au bout de 55 secondes, on lit :

T=2,414062501.

$(3-2)/2^{11}$ est égal à 5.10^{-4} environ ; la vraie valeur de T ($1+\sqrt{2}$ = 2,414213562) est bien à moins de 5.10^{-4} de la valeur affichée. Après ENTER, on lit :

MAX F = 1,17157286,

ce qui est excellent ($4-2\sqrt{2}$ = 1,171572876).

Commentaires sur le programme : le test de la ligne 60 correspond à la question f(B)>f(C) ? S'il est positif, A ne change pas, et le nouveau E est

Essayez de tirer un maximum

```
001   LBL A
002   ST I      n
003   R↓
004   STO 4     b(=E)
005   R↓
006   STO 0     a(=A)

007   LBL E
008   RCL 0 ┐
009   RCL 4 │
010     +   │  Calcul
011     2   │  de C
012     ÷   │
013   STO 2 ┘
014   RCL 4 ┐
015     +   │
016     2   │  Calcul de D
017     ÷   │
018   STO 3 ┘
019   RCL 0 ┐
020   RCL 2 │
021     +   │  Calcul
022     2   │  de B
023     ÷   │
024   STO 1 ┘
025   GSB B     f(B)
026   STO 5
027   RCL 2
028   GSB B     f(C)
029   STO 6
030   RCL 5
031    x>y      f(B)>f(C) ?
032   GTO C
033   RCL 3
034   GSB B     f(D)
035   RCL 6
036    x≤y      f(C)≤f(D) ?
037   GTO D
038   RCL 1 ┐
039   STO 0 │  B est le nouvel A,
040   RCL 3 │  D est le nouvel E
041   STO 4 ┘
042   LBL 0
043    DSZ      (compteur)
044   GTO E
045   RCL 0 ┐
046   RCL 4 │
047     +   │  Calcul de T
048     2   │
049     ÷   ┘
050    R/S      (lecture de T)
051   GSB B     f(T)
052    RTN      (lecture de M)

053   LBL C
054   RCL 2 ┐  C'est le nouvel
055   STO 4 ┘  E
056   GTO 0

057   LBL D
058   RCL 2    C'est le nouvel
059   STO 0    A
060   GTO 0

061   LBL B ┐  Mettre ici le
062    RTN  ┘  calcul de f(x)
```

l'ancien C ; en d'autres termes on remplace [A,E] par [A,C].

Le test de la ligne 80 — qui n'est mis en œuvre que si f(B)≤f(C) — correspond a f(C)<f(D) ? S'il est positif, A est changé en C, E reste invariant, [A,E] devient [C,E]. Enfin, dans les autres cas, [A,E] est remplacé par [B,D], ce qui correspond bien à l'organigramme.
Le BEEP est destiné aux distraits.

Programme sur HP 67 (peut être transposé sur d'autres HP ou TI)

Mode opératoire : mettre le calcul de f à partir du pas n° 62 ; ici, on inscrira par exemple :

```
061   LBL B
062   STO 7     x
063     6
064     −       x−6
065   RCL 7
066     ×       x²−6x
067     7
068     +       x²−6x+7
069     1
070   RCL 7
071     −       1−x
072     ÷       f(x)
073    RTN      (ex 062)
```

Passer en mode RUN et faire : a ENTER b ENTER n A (par exemple ici 2 ENTER 3 ENTER 10 A). Au bout de 40 secondes, on lit T, soit : 2,414062501 (erreur : 15.10^{-5}). Après R/S, on lit M, soit : 1,171572859 (erreur : 17.10^{-9}).

Les résultats sont les mêmes sur chaque machine, et ils sont excellents (surtout pour M).

Commentaires sur le programme : la HP 67 n'ayant pas le test x<y, on a mis x≤y (pas 036) ce qui n'a aucune espèce d'importance comme on pourra s'en convaincre. Pour le reste, ce programme est une réplique fidèle de celui du PC 1211. Le calcul de f(x) se fait par le sous-programme B.

Contenus des mémoires :

0	1	2	3	4	5	6
A	B	C	D	E	f(B)	f(C)

Rappelons que le DSZ (Decretement and Skip if Zero) diminue d'une unité le contenu de la mémoire indirecte I (où l'on a stocké l'entier n) ; au bout de n opérations élémentaires, ce contenu devient nul et, au lieu de recommencer à exécuter le label E, on saute à l'instruction 045 qui débouche sur l'affichage final.

Encore un mot sur la nécessité, signalée plus haut, d'être modeste dans le choix de n ; avec n=20 (d'où, théoriquement, une précision égale à $b-a/2^{n+1}=1/2^{21}=5.10^{-7}$) on trouve :
Sharp (110 sec) T = 2,414215092
MAX F = 1,171572876
Hewlett (85 sec) T = 2,414244654
MAX F = 1,171572875

Pour la Sharp, M est exact ; l'erreur sur T est de 15.10^{-7} (et non pas 5.10^{-7}). Avec HP, on obtient respectivement des erreurs égales à 10^{-9} et 300.10^{-7} (soixante fois trop forte).

Cette méthode peut être utilisée pour résoudre une équation du type f(x)=0, en recherchant le maximum de $-|f(x)|$ (ou de $-[f(x)]^2$). Dans l'exemple étudié, il y a une racine entre 1 et 2, égale à $3-\sqrt{2}$ = 1,585786438.

Avec la Sharp, écrivons 150 :
X=−ABS ((XΛ2−6*X+7)/(1−X))

Pour a=1, b=2, n=10, on trouve en 55 secondes : T = 1,585937501 (erreur : $1,5.10^{-4} < 3.10^{-4}$), MAX F = -7.10^{-4} (au lieu de 0).

Une Hewlett, avec (073 ABS/074 CHS/075 RTN) donne, en 45 secondes, les mêmes valeurs de T et de M. La méthode est donc utilisable. Evidemment, il faut être certain que f est strictement monotone sur [a,b], et y possède une racine.

Voici un autre exemple, appliqué à f(x) = cos(x/2) avec a=3, b=4. Pour n=10 on trouve, au lieu de π, le nombre 3,141601563, soit une erreur de 9.10^{-6} ($<5.10^{-4}$).

Employées sur des intervalles a priori non adéquats (c'est-à-dire si l'on recherche un maximum et qu'il peut y en avoir plusieurs ou qu'il existe des minimums), ces méthodes donneront quand même toujours un maximum local ou une racine, s'il en existe.

☐ André Warusfel

Un pot commun pour toutes les machines

La TI 57 relève le gant

Dans notre précédent numéro, nous avons prétendu qu'il était " sans doute " impossible de faire entrer le programme d'élévation à la puissance cinquième d'une matrice 2×2 dans une TI 57. Il n'en fallait pas plus pour que vous nous apportiez la preuve du contraire : vous avez relevé le défi.
Bravo !

```
      MATRICE
  AUTEUR : YVES ROQUE
  COPYRIGHT L'ORDINATEUR
    DE POCHE ET L'AUTEUR
 *************************
 0   38  0     2ND EXC  0
 1   75        +
 2   33  2     RCL  2
 3   38  1     2ND EXC  1
 4   39  1     2ND PRD  1
 5   38  6     2ND EXC  6
 6   39  6     2ND PRD  6
 7   38  6     2ND EXC  6
 8   43        (
 9   38  5     2ND EXC  5
 10  55        MULTIPLIE PAR
 11  38  4     2ND EXC  4
 12  39  4     2ND PRD  4
 13  38  4     2ND EXC  4
 14  34  1     SUM  1
 15  33  0     RCL  0
 16  44        )
 17  34  5     SUM  5
 18  33  2     RCL  2
 19  38  3     2ND EXC  3
 20  39  3     2ND PRD  3
 21  38  6     2ND EXC  6
 22  39  6     2ND PRD  6
 23  38  6     2ND EXC  6
 24  43        (
 25  22        X ECHANGE T
 26  55        MULTIPLIE PAR
 27  38  4     2ND EXC  4
 28  39  4     2ND PRD  4
 29  38  4     2ND EXC  4
 30  34  3     SUM  3
 31  33  0     RCL  0
 32  44        )
 33  34  7     SUM  7
 34  00        0
 35  85        =
 36  38  0     2ND EXC  0
 37  56        2ND DSZ
 38  71        RST
 39  33  1     RCL  1
 40  81        R/S
 41  33  3     RCL  3
 42  81        R/S
 43  33  5     RCL  5
 44  81        R/S
 45  33  7     RCL  7
 46  81        R/S
```

■ Voici un petit programme (47 pas) qui vous montrera qu'on peut élever une matrice 2×2 à la puissance n à l'aide d'une TI 57. Il y a mêmes les " coups de cymbale " pour sortir les résultats ! On remarquera que ce programme est en fait la répétition de 2 parties (pas 02 à 17, et pas 18 à 33) où seules les mémoires de numéro impair changent de contenu.

Dès le départ, il y a 9 chiffres à conserver et nous ne disposons que de 8 mémoires, c'est pourquoi j'en ai stocké un (n) dans le registre x. Par la suite (pas 10 et 26), ce nombre passe dans le registre y pour faire place à un autre : c, puis d. A la fin du programme d' (la mémoire 0) est perdu.

Il ne faut pas oublier que c'est n − 1 et non pas n qu'il faut inscrire avant le calcul dans la mémoire O. Ainsi pour obtenir le carré d'une matrice, on exécutera 1 STO 0, pour le cube 2 STO 0, etc.

☐ Yves Roque

Exemple d'exécution :

$$\begin{bmatrix} 15 & 81 \\ 9 & 64 \end{bmatrix}^4 = \begin{bmatrix} 5459805 & 36979821 \\ 4108869 & 27830314 \end{bmatrix}$$

Faire 15	STO 1,	STO 2	(a)
9	STO 3,	STO 4	(b)
81	STO 5,	STO 6	(c)
3	STO 0,		(n − 1)
64	STO 7		d
	RST		
	R/S		

Affichage de e.
Faire d'autres R/S pour afficher f, g et h.

Un petit 421 pour HP 33 et 34

■ Voici, en 39 pas de programme, un 421 pour HP33 et 34. Avant de jouer, on donne au générateur aléatoire un nombre-source compris entre 0 et 1 et l'on presse sur R/S. On peut alors lancer trois dés, deux seulement d'entre eux ou même un seul en précisant le(s)quel(s). Le petit tableau ci-dessous vous indique comment choisir votre jet de dés.

01	GSB 29	21	GSB 29
02	STO 2	22	STO 3
03	GSB 29	23	GTO 07
04	STO 3	24	GSB 29
05	GSB 29	25	STO 2
06	STO 4	26	GSB 29
07	RCL 2	27	STO 4
08	PAUSE	28	GTO 07
09	RCL 3	29	RCL 0
10	PAUSE	30	9
11	RCL 4	31	9
12	GTO 00	32	7
13	GSB 29	33	X
14	STO 2	34	FRAC
15	GSB 29	35	STO 0
16	STO 3	36	6
17	GTO 07	37	X
18	GSB 29	38	1
19	STO 2	39	+
20	GTO 07	40	INT
		41	RTN

Dés à jouer :	n°1	n°2	n°3
GSB 01	oui	oui	oui
GSB 03	non	oui	oui
GSB 05	non	non	oui
GSB 13	oui	oui	non
GSB 18	oui	non	non
GSB 21	non	oui	non
GSB 24	oui	non	oui

Par exemple, avec GSB 01, un joueur obtient 5-4-3 à son premier lancer. Il veut rejouer les dés n°1 et 3. Il demande donc GSB 24. Il obtient alors par exemple 1-4-6 (le deuxième dé n'a pas été relancé). Il veut rejouer le dé n°3 : GSB 05. S'il a de la chance, il gagne avec 1-4-2 !

☐ Jean-Christophe Janin

Un pot commun pour toutes les machines

Le nombre secret de la TI 57

☐ Dans cette version du *Nombre secret,* le nombre que la machine vous invite à deviner est compris entre 0 et 100. Ce nombre aléatoire est " fabriqué " dans la petite loterie mathématique du sous-programme Lbl 0 aux pas 33 à 46. Comme sur les autres ordinateurs de poche, le jeu consiste à deviner quel est le nombre secret en proposant une série de fourchettes (50-80 par exemple). A chaque essai, la machine indique si la fourchette

. contient le nombre à deviner : affichage **2**

. est trop basse : affichage **−1**

```
      NBR SECRET
  AUTEUR : BRUNO GARREAU
  COPYRIGHT L'ORDINATEUR
    DE POCHE ET L'AUTEUR
*************************
 0   81        R/S
 1   32  1     STO  1
 2   32  7     STO  7
 3   81        R/S
 4   32  2     STO  2
 5   66        2ND X = T ?
 6   51  1     GTO  1
 7   -76       INV 2ND X >= T ?
 8   51  2     GTO  2
 9   33  4     RCL  4
10   22        X ECHANGE T
11   76        2ND X>=T
12   51  3     GTO  3
13   33  2     RCL  2
14   -76       INV 2ND X >= T ?
15   51  4     GTO  4
16   02        2
17   71        RST
18   86  1     2ND LBL  1
19   33  4     RCL  4
20   -66       INV 2ND X=T ?
21   03        3
22   51  9     GTO  9
23   86  2     2ND LBL  2
24   15        CLR
25   71        RST
26   86  4     2ND LBL  4
27   01        1
28   84        +/-
29   71        RST
30   86  3     2ND LBL  3
31   01        1
32   71        RST
33   86  0     2ND LBL  0
34   55        MULTIPLIE PAR
35   06        6
36   07        7
37   85        =
38   -49       INV 2ND INT
39   55        MULTIPLIE PAR
40   02        2
41   -18       INV 2ND LOG
42   85        =
43   49        2ND INT
44   32  4     STO  4
45   15        CLR
46   71        RST
```

. est trop haute : affichage **1**

. ou si elle n'a pas de sens (le joueur a proposé, par distraction ou par malice, une fourchette dont la borne haute est plus petite que la borne basse), l'affichage indique alors 0.

Le joueur propose autant de fourchettes qu'il le désire jusqu'au moment où il pense avoir découvert la solution. Il ne lui reste plus qu'à proposer une " fourchette " dont les deux extrémités sont identiques (il demande en quelque sorte : " le nombre est-il compris entre 23 et 23 ? ") ; la machine lui répond 3 s'il s'est trompé et fait clignoter la bonne solution à l'affichage s'il a deviné juste.

Après avoir déclenché la génération du nombre au hasard (afficher un nombre compris entre 0 et 1 et faire SBR 0), le joueur introduit au clavier la borne basse de sa fourchette. En appuyant sur R/S (pas 00), il conduit le programme à engranger ce nombre dans les mémoires 1 et 7 (pas 01 et 02). Il introduit alors la borne haute qui est stockée dans la mémoire 2 au pas 04. Le programme examine ensuite si ces deux bornes sont égales : x = t ? du pas 05 (1). Si c'est bien le cas, c'est que le joueur a proposé une solution : il pense avoir deviné le nombre secret et le programme se poursuit au sous-programme Lbl 1 (GTO 1 du pas 06) où l'on regarde si cette solution est différente de la bonne (RCL 4, x ≠ t des pas 19 et 20). Si la solution est différente, c'est le pas 21 qui est exécuté et l'affichage indique 3 : le joueur s'est trompé. Sinon, on passe au pas 22 où l'instruction impossible GTO 9 provoque le clignotement de l'affichage : le joueur a gagné.

Au pas 07, on regarde si la borne haute qui se trouve à l'affichage (registre x) est plus petite que la borne basse, auquel cas le joueur a proposé une fourchette impossible et le programme se poursuit à l'étiquette 2 (GTO 2 du pas 08) ; on efface l'affichage et l'on redemande une fourchette : CLR et RST. Si la fourchette est plausible, on place le nombre secret dans le registre 7 (RCL 4 et x >< t des pas 10 et 11) et l'on compare la borne basse (à l'affichage) et le nombre secret : x ⩾ t. Si la borne basse est supérieure **ou égale** à la bonne solution, on se rend au pas 30 (GTO 3) pour afficher 1 et l'on retourne au début du programme pour l'introduction d'une nouvelle fourchette. Dans le cas contraire, on rappelle la borne haute à l'affichage (RCL 2 du pas 13) et l'on examine si elle est plus petite que la solution, x < t ? du pas 14. La borne haute est-elle bien plus petite que la solution ? Si oui, c'est le pas 15 qui est exécuté et l'on passe donc au sous-programme 4 des pas 26 à 29 qui affichent −1 et renvoient au début du programme.

Lorsque la réponse à tous ces tests a été négative, on est certain que le nombre secret est compris à l'intérieur de la fourchette : les pas 16 et 17 affichent 2 et renvoient en 00 pour l'introduction d'une nouvelle fourchette.

☐ Bruno Garreau

(1) Sur le fonctionnement des tests de la TI 57, le lecteur pourra se reporter aux explications qui sont fournies dans l'article des pages 45 à 49.

Utilisation du programme

1 - afficher un nombre quelconque compris entre 0 et 1 et faire SBR 0

2 - introduire la limite inférieure de la fourchette et presser sur R/S.

3 - introduire la limite supérieure de la fourchette et presser sur R/S.

4 - 0 à l'affichage : la fourchette n'a pas de sens, on appuie sur CLR et on recommence à l'étape n° 2.

1 à l'affichage : la fourchette est trop haute, on appuie sur CLR et l'on recommence en 2.

−1 à l'affichage : la fourchette est trop basse, on appuie sur CLR et on repart en 2.

3 à l'affichage : la solution proposée était fausse. On appuie sur R/S et l'on est en 2. La solution que l'on a proposée clignote à l'affichage : on a découvert le nombre secret.

Pour une autre partie, reprendre à l'étape n° 1.

NOUVEAU

NOUVEAU

La découverte du PC-1211
par Jean-Pierre Richard

Au fil des chapitres, cet ouvrage fera découvrir, au non initié, instructions et commandes, variables et mémoires, fonctions périphériques. Enrichi d'exercices d'applications et d'un index, ce manuel fournit à l'utilisateur tous les éléments de base nécessaires à la programmation en langage Basic du PC-1211 (ou TRS-pocket).

Série verte
152 Pages - 75 FF

La découverte de la TI-57
par Xavier de la Tullaye

S'adressant aux débutants, cet ouvrage les conduira, dans un langage clair, de l'élémentaire 2 + 2 à des programmes perfectionnés. Après une étude fonctionnelle de la calculette, la programmation est expliquée progressivement, de la conception à la réalisation en s'appuyant sur de nombreux exemples.

Série verte
144 Pages - 65 FF

Programmer en L.S.E.
par Stéphane Berche et Yves Noyelle

Cet ouvrage donne tous les éléments nécessaires pour l'utilisation d'un ordinateur programmable en LSE (langage français). L'ensemble exposé forme un tout cohérent permettant de s'initier au langage. Il est complété d'exemples pratiques et de plusieurs programmes opérationnels.

Série verte
128 Pages - 65 FF

Programmer en Basic
par Michel Plouin

Ce livre a été écrit pour les utilisateurs d'ordinateurs individuels en particulier d'Apple II, TRS-80 et PET/CBM. Un répertoire Basic rend son utilisation très pratique et facilite la transposition d'un programme écrit pour un P.S.I. sur un autre.

Série verte
132 Pages - 65 FF

Programmer en assembleur
par Alain Pinaud

Cet ouvrage constitue une introduction complète au langage machine, et à son frère l'assembleur, comprenant des exercices et des exemples. Bien qu'illustré par le code du Z80, il sera d'une lecture tout aussi utile aux possesseurs de P.S.I. disposant d'un autre microprocesseur.

Série bleue
144 Pages - 75 FF

Editions du P.S.I.
41-51, rue Jacquard
BP 86 - 77400 Lagny-s/Marne
Téléphone (6) 007.59.31

au Canada : SCE Inc.
3449 rue Saint-Denis
Montréal Québec H2X3L1
Tél. : (514) 843.76.63

BON DE COMMANDE

Envoyer ce bon accompagné de votre règlement à
EDITIONS DU P.S.I.
41-51, rue Jacquard BP 86
77400 Lagny-s/Marne
Tél. (6) 007.59.31

DESIGNATION	NOMBRE	PRIX
	TOTAL	

Les prix sont : taxes, emballage et port compris.
(par avion : ajouter 5 FF par livre)

NOM ..... PRENOM .....

rue ..... N° .....

Code post. |_|_|_|_|_| Ville .....

OP 3

un ancêtre...

Compucorp Scientist

Le compucorp aura bientôt dix ans, presque une antiquité... mais on s'en souvient avec les yeux du cœur : quelle belle machine !

■ La marque Compucorp a fait son apparition en France entre 1972 et 1975, représentée par I.C.E.

Deux modèles de calculatrices programmables figuraient au SICOB 72 : la « Scientist » présentée ici et la « Statistician ». La Scientist n'offrait pour sa première apparition que 80 pas de programme, et elle fut remplacée en 1973 par une version comportant 2 fois 80 pas.

N'ayant eu ces calculatrices qu'à titre de prêt en octobre 1972, je ne peux plus donner que des caractéristiques physiques approximatives : 23 × 14 × (5) cm ; poids : environ 1,4 kg ; petite mallette de transport. Batterie rechargeable ; mémoire volatile.

Arithmétique. Clavier standard : 0 à 9, Exp, Chg sign, et Clear entry ; set D.P. ("decimal point") 0 à 9.
Affichage à 10 chiffres + exposant à 2 chiffres par diodes électroluminescentes ; calculs, résultats et stockage à 13 chiffres ; π à 13 chiffres. Opérateurs arithmétiques : + − × ÷ = (sans priorité) ; 10 registres mémoire (0 à 9) ; accès par RCLn, STOn et Exchn.

Fonctions. Les fonctions classiques y figurent : $\sqrt{x}$, 1/x, sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, Ln, LOG, e^x, 10^x, y^x (notée a^x) ; fonctionnement en degrés ou grades par permutateur ; conversions Rectangulaire <=> Polaire, Degré <=> Radian, Décimal <=> Sexagésimal.
Observation : la lacune la plus gênante est celle de INT.

Programmation. Permutateur RUN/LOAD, touche START/STOP, permutateur programme 1/2 (avec départ au pas 1).
Ni test, ni branchement, ni SST, ni BST ; 1 pas de programme par touche pressée.

Fonctionnement. Pile à 4 niveaux autorisant 2 parenthèses. Une particularité : non seulement, comme chez TI, les opérations « blanches » diverses effectuées par le registre d'entrée (X) ne déplace pas le contenu du registre Y, mais celui-ci sert également de stockage de constante tant qu'un opérateur arithmétique ou une fonction à 2 entrées ne copie pas le contenu du registre d'entrée dans le registre Y ; cette disposition est fort agréable pour certains types de calculs :

K × A = affichage de KA
B = affichage de KB
C = affichage de KC, etc.

Les opérateurs arithmétiques + − × ÷ comme les fonctions à 2 entrées provoquent l'exécution de l'opération en place et le résultat est transféré dans le registre Y. Outre la particularité ci-dessus, les opérations entre registre X et mémoire peuvent s'effectuer dans les 2 sens, c'est-à-dire non seulement STOn STO + n,... STO÷n, mais aussi RCL + n, ... RCL÷n, possibilité que je n'ai plus retrouvée sur d'autres calculatrices (et qui serait pourtant fort utile sur HP par exemple).

Abordons la fonction de la dernière touche : 2nd FUNC. qui apportera la réponse à une question qu'ont pu se poser les lecteurs à la vue du clavier. En effet, les fonctions SIN et COS, $\sin^{-1}$ et $\cos^{-1}$, °→R et R→°, LN et LOG, e^x et 10^x figurent sur la même touche ; les 2 fonctions sont calculées simultanément, la première nommée est placée dans le registre et à l'affichage, la seconde dans un registre particulier où elle subsiste jusqu'à l'exécution de la fonction suivante utilisant ce registre ; la permutation des valeurs se fait par cette touche 2nd FUNC. Il en est de même pour les fonctions P→R (X à l'affichage) et R→P (θ à l'affichage) (1).

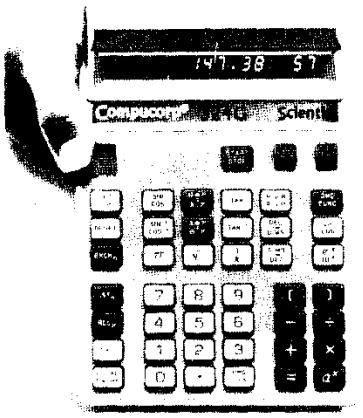
La grande difficulté pour la programmation de ces calculatrices provenait du fait que la position LOAD n'enregistrait pas le programme « à blanc ». Autrement dit, la calculatrice effectuait à mesure ce qu'on pourrait appeler l'exemple-type : il fallait donc que cet exemple ne provoque pas d'erreur au cours de sa réalisation, ce qui n'était pas toujours évident lorsque le programme devait s'arrêter sur l'instruction START/STOP pour l'introduction de données. En outre, autant que je me souvienne, il n'était pas possible de contrôler et encore moins de corriger le programme introduit : une erreur obligeait donc à reprendre la programmation entièrement.

Exemple de décision logique extraite d'un programme : sur une entrée X comprise entre −360° et 360°, exprimer l'angle X de 0 à 360°

Sur Compucorp Scientist :

(X)	(7 pas)
+	la parenthèse suivante évite que R→P ne provoque X + Y =
(	pas besoin de valider l'entrée comme sur TI
R→P	
0	si X<0 θ = 180, si X≥0 θ = 0
+	fonction calculée : θ à l'affichage et recopié en Y
(	résultat : 0 ou 360
=	X entre 0 et 360°

ce qui correspond à la solution
HP 41 : 360 MOD (4 octets)
TI 59 + x ⇄ t 0 GE (4 pas) 360 = (10 pas).



L'absence de branchements, de décisions logiques, etc. posait également quelques problèmes. Toutefois, pour les décisions logiques, la fonction R→P était utilisable, car elle possédait le seul test interne n'entraînant pas « erreur » : en effet pour x = 0 et y = 0, cette fonction donnait r = 0 et θ = 0. On pouvait ainsi programmer : prendre le plus grand (petit) de 2 nombres, signe de x, valeurs discrètes...

Malgré ses capacités mémoire réduites et son faible nombre de pas de programme, j'ai pu programmer sur la calculatrice à 2 programmes la résolution de 4 équations simultanées (méthode par élimination, bien entendu, avec 10 mémoires). Le programme n° 1 permettait d'entrer et de traiter les coefficients des variables, le programme n° 2 entrant le terme constant de chaque équation et donnant en fin les valeurs des variables. Les équations étaient entrées une par une avec passage du programme 1 au programme 2 pour entrer la constante, puis retour au programme 1 pour l'équation suivante, etc. Les lecteurs souriront sans doute à l'évocation de ces manipulations, mais cette calculatrice était bien (à ma connaissance) la première et la seule programmable de cette taille à pouvoir traiter ce problème.

Son agrément était une rapidité de calcul assez supérieure à celle des calculatrices de la génération suivante.

Jean Thiberge

Imprimerie SIMA, 114-116 Av. Pierre Brossolette, 92240 Malakoff. Printed in France. Directeur de la Publication : Jean-Luc Verhoye
Dépôt légal 3e trimestre 1981. Diffusion NMPP. Numéro de Commission paritaire : 63617

UN DOCUMENT ESSENTIEL

Avant d'acheter votre ordinateur lisez le

GUIDE 81-82

de L'ORDINATEUR INDIVIDUEL

au sommaire :

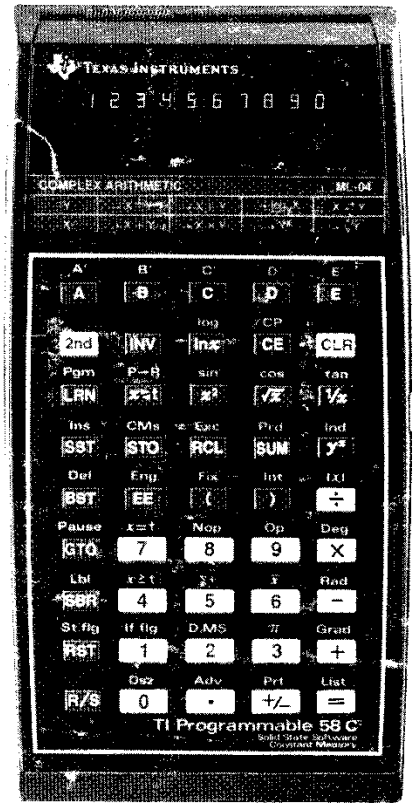
- **Panorama des ordinateurs valant entre 250 FF et 60 000 FF (plus de 80 matériels).**
- **Panorama des imprimantes valant moins de 10 000 FF (plus de 30 matériels).**
- **Réactualisation de 22 bancs d'essai parus dans l'OI depuis le N° 1.**
- **Annuaire des fournisseurs (plus de 500 adresses).**
- **Annuaire des clubs (plus de 200 adresses).**
- **Dictionnaire de l'informatique individuelle.**
- **Le point sur les nouveautés parues depuis l'été 1980.**
- **Et une série d'articles pour vous "guider" sur le chemin de votre informatique individuelle.**

25 FF* chez votre marchand de journaux

Pour recevoir, chez vous, le Guide 81-82,
il vous suffit d'envoyer vos nom et adresse ainsi qu'un chèque de 25 FF* à
L'ORDINATEUR INDIVIDUEL **(GUIDE 81-82)** 41, rue de la Grange aux Belles - 75483 Paris Cedex 10

* Etranger 30 FF

Imaginez, la TI-58 C fera le reste.



Chez Texas Instruments, nous pensons que pour mieux tirer profit de l'imagination, il lui faut réduire toute la partie répétitive des calculs.

Et avec la TI-58 C, le problème est résolu ! C'est une calculatrice spécialement conçue pour réaliser les calculs les plus complexes et surtout, pour suivre le fil de vos pensées...Dans les domaines les plus variés.

En effet, la TI-58 C possède 172 fonctions, 60 mémoires non volatiles, 480 pas de programme, directement accessibles au clavier.

Et ce n'est pas tout... La TI-58 C est équipée d'un module de base de 25 programmes (traitements des matrices, statistiques, finances, mathématiques). Vous pouvez multiplier encore cette puissance par une bibliothèque de 10 modules spécialisés et programmés. Soit 5 000 pas de programme par module.

C'est le nec plus ultra à un prix étonnamment bas !

Et si la TI-58 C s'adapte sur mesure aux travaux des étudiants et des professionnels, vous pouvez aussi utiliser sa grande capacité de mémoire, pour jouer sans compter.

Alors laissez aller votre imagination, Texas Instruments fera le reste !

Le progrès qui fait progresser.

TEXAS INSTRUMENTS